

# Roadmap für die Migration von REAL-OQS v0.0605 nach CNNA

Complement-Netzarchitektur als gemeinsame Stammtheorie für mehrere QFT-Faces

Dokumenttyp:	architekturgebundene Migrationsroadmap
Codebindung:	Archiv REAL-OQS_v0.0605_full_no_comments.zip
Interne Leitdokumente:	CNNA-Spezifikation v0.1; Lean-Dokumentationsregelwerk v1.8
Externe Leitquellen:	Snowmass-White-Paper zur QFT-Definition; AQFT/LCQFT/FFT/CFT/AS-, Defekt-, Anomalie- und String-Mathematik-Literatur
Status:	keine Phasenplanung, sondern Pfeilerweise Vorgehensweise mit Abhängigkeiten, Gates und Stop-Regeln
Datum:	02. April 2026
Editor:	Jan Seeck (antaris)
Ghostwriter:	ChatGPT

Diese Fassung ist *regelwerksbasiert*: Sie ist explizit an ein Archiv gebunden, trennt zwischen portierbarem Kern, neu zu schreibenden Modulen, Bright-Spezialfall und späten Seeds/Windows, und organisiert die Migration nicht als Zeitplan, sondern als partielle Ordnung fachlicher Arbeitspakete. Die Motivation für eine mehrgesichtige CNNA-Stammtheorie folgt sowohl aus der internen Spezifikation als auch aus der in der Snowmass-Übersicht betonten Tatsache, dass keine einzelne heute etablierte Axiomatik alle physikalisch relevanten QFT-Situationen abdeckt; deshalb ist das Ziel nicht *eine* einzig privilegierte Oberflächenbeschreibung, sondern eine gemeinsame tiefere Struktur, aus der mehrere QFT-Faces sektor- und regimeabhängig emergieren.[4, Abstract; pp. 1, 4–6, 9–12][2, pp. 1–10, 20–36][3, pp. 7–8]

## Inhaltsverzeichnis

<b>Dokument-Metadaten</b>	<b>4</b>
<b>1 Einleitung</b>	<b>4</b>
<b>2 Ziel und Geltungsbereich</b>	<b>6</b>
2.1 Ziel . . . . .	6
2.2 Geltungsbereich . . . . .	6
2.3 Was dieses Dokument nicht behauptet . . . . .	6
<b>3 Dokumenttyp, Quellenhierarchie und Minimal-Audit</b>	<b>7</b>
3.1 Quellenhierarchie . . . . .	7
3.2 Reifegrad-Konvention und Abbildung auf das Regelwerk . . . . .	7
3.3 Zwang zur lesbaren Notation für alle Lean-Objekte . . . . .	7
3.4 Warum eine Stammtheorie mit mehreren Faces? . . . . .	8
3.5 Aktuelle Größenordnung des inspectierten Archivs . . . . .	8
3.6 Roadmap als lebendes Gate-Dokument: prototypisches Vorseilen und Rückfluss von Erkenntnissen . . . . .	9
3.7 Mehrstufiges Wahrheitsprotokoll: AI, Formalisierung, Verifikation, Semantik und Physik . . . . .	10
3.8 Operative Verdichtung des aktiven Migrationspfads . . . . .	11
<b>4 Definitionskatalog der Kernobjekte</b>	<b>13</b>
4.1 Formaler Definitionskatalog des aktiven und spät aktivierten Pfads . . . . .	13
4.2 Notationspflicht innerhalb des Definitionskatalogs . . . . .	13
<b>5 Projektglossar und Verweißstruktur</b>	<b>14</b>
5.1 Kernbegriffe des Projekts . . . . .	14
5.2 Strikte Strukturkarte . . . . .	15
<b>6 Pillar-Handoff-Spezifikationen</b>	<b>16</b>
6.1 Handoff $A \rightarrow B$ . . . . .	16
6.2 Handoff $B \rightarrow C$ . . . . .	16
6.3 Handoff $C \rightarrow D$ . . . . .	16
6.4 Handoff $D \rightarrow E$ . . . . .	16
<b>7 Migrationsgesetz: was direkt portiert, was umgeschrieben und was dezentriert wird</b>	<b>18</b>
7.1 Direkt oder nahezu direkt portierbarer Kern . . . . .	18
7.2 Module, die semantisch neu gelesen werden müssen . . . . .	18
7.3 Was nur noch als Bright-Spezialfall dienen darf . . . . .	18
7.4 Späte Seeds und diagnostische Inputs für D und E . . . . .	19
<b>8 Pillar A – ontische Stammtheorie und Komplementarchitektur</b>	<b>20</b>
8.1 Zielkern von A . . . . .	20
8.2 Konkrete Vorgehensweise in A . . . . .	20
8.3 Abnahmekriterien für A . . . . .	22
8.4 Was in A strikt zu vermeiden ist . . . . .	22

<b>9</b>	<b>Pillar B – AQFT aus dem Komplementnetz statt aus dem Bright-Rand</b>	<b>23</b>
9.1	Zielkern von B . . . . .	23
9.2	Konkrete Vorgehensweise in B . . . . .	23
9.3	B-Kernformalismus der modularen Darstellungsarchitektur . . . . .	24
9.4	Warum das Komplementnetzwerk für full-derived lokale QFT physikalisch zwingend wird . . . . .	25
9.5	Zwingende AQFT-Modulergänzungen im Modulbaum . . . . .	26
9.6	Abnahmekriterien für B . . . . .	27
9.7	String-mathematische Ideen, die B vorbereiten muss . . . . .	28
<b>10</b>	<b>Pillar C – offene Quantensysteme, Kanäle und Rückwirkung</b>	<b>29</b>
10.1	Zielkern von C . . . . .	29
10.2	Konkrete Vorgehensweise in C . . . . .	29
10.3	Abnahmekriterien für C . . . . .	30
10.4	Was in C nicht passieren darf . . . . .	30
<b>11</b>	<b>Pillar D – emergente Geometrie, lokale Kovarianz und Gravity-Routen</b>	<b>31</b>
11.1	Zielkern von D . . . . .	31
11.2	Konkrete Vorgehensweise in D . . . . .	31
11.3	Abnahmekriterien für D . . . . .	33
11.4	Minimaler D-Dateibaum und explizite D/E-Gate-Reihenfolge . . . . .	33
11.5	Theoretische Rahmung für D . . . . .	34
<b>12</b>	<b>Pillar E – später Gauge/Matter/SM-Strang aus der Stammtheorie</b>	<b>36</b>
12.1	Zielkern von E . . . . .	36
12.2	Feynman-Audit für Matter: was schon vor E als Gate relevant ist . . . . .	36
12.3	Feynman-Audit-Matrix für die Planung von D/E . . . . .	37
12.4	Konkrete Vorgehensweise in E . . . . .	37
12.5	Minimaler Zieldateibaum für E und seine Vorzone . . . . .	39
12.6	Abnahmekriterien für E . . . . .	40
12.7	Späte Horizontschicht für Teilchen-, Scattering- und IR-Physik . . . . .	41
12.8	Was die String-Mathematik hier genau liefert . . . . .	41
<b>13</b>	<b>Pfeilerübergreifende partielle Ordnung statt Phasenplan</b>	<b>42</b>
13.1	Gate-Reife statt Parallelillusion . . . . .	44
13.2	Operativer Unterpfad 6a–6f für den modularen B-Kern . . . . .	44
13.3	Spät-B-/Früh-E-Ergänzung für Nuclearity, DHR/DR und physikalische Rekonstruktion . . . . .	44
13.4	Operativer Unterpfad PC-a–PC-d für Parameter-Closure zwischen C und D . . . . .	45
13.5	Drei harte Zusatzregeln für Abhängigkeiten und Gates . . . . .	45
13.6	Neue Gates für den modularen B-Kern und die B->D-Brücke . . . . .	45
13.7	Weitere Gates für physikalische Vollständigkeit jenseits des heutigen Kerns . . . . .	46
<b>14</b>	<b>Datei-/Kettenblätter für tragende Pfade</b>	<b>47</b>
14.1	Kettenblatt des modularen B-Kerns . . . . .	47
14.2	Kettenblatt des Parameter-Closure-Strangs . . . . .	47
14.3	Kettenblatt der D/E-Brücke . . . . .	48
<b>15</b>	<b>Operative Regeln für De-seeding, Regression und Benennung</b>	<b>49</b>
15.1	De-seeding-Regel . . . . .	49
15.2	Regression gegen den Altpfad . . . . .	49
15.3	Benennungsregel und Immediate Readability Policy . . . . .	49
15.4	Konsistente Ableitung der Notation-Dateien aus der Roadmap . . . . .	49

---

15.5 Repo-Infrastruktur, Importgraph und Rollback . . . . .	50
<b>16 Schlussurteil</b>	<b>51</b>
<b>A Erweiterte Modulmatrix für die Migration</b>	<b>52</b>
A.1 Direkte oder nahezu direkte Ports . . . . .	52
A.2 Semantische Umschriften . . . . .	52
A.3 Bright-Spezialfall und Regression . . . . .	52
<b>B Vollständige Disposition des BoundaryMatrix-Derived-Blocks</b>	<b>54</b>
<b>C Weitere Pillar-A Module aus Core, Symmetry und Kinematics</b>	<b>55</b>
<b>D Disposition des B-Gates- und Meta-Subsystems</b>	<b>56</b>
D.1 B-Gates als eigene Validierungsschicht . . . . .	56
D.2 Meta-Dateien als Regressions- und Auditstrang . . . . .	56
<b>E Quell-Ziel-Abgleich für Namens- und Topologiefragen</b>	<b>57</b>
E.1 Notationsmodule als Zielarchitektur . . . . .	57
E.2 Notations- und Alias-Appendix für den aktiven CNNA-Pfad . . . . .	57
<b>F Variablen-, Namens- und Notationsregister</b>	<b>67</b>
F.1 Register I: Objekt- und Aliasregister . . . . .	67
F.2 Register II: Symbol- und Parameterregister . . . . .	77
F.3 Register III: Reifegrad- und Architekturstatusregister . . . . .	79
F.4 Register IV: Meta-Methodikregister . . . . .	81
<b>G Instanz- und Kontextblatt für tragende Pfade</b>	<b>84</b>
<b>H Prüfprotokoll der Roadmap</b>	<b>85</b>
<b>I Namens- und Pfadverzeichnis</b>	<b>86</b>
<b>J Meta-Audit-Marker</b>	<b>87</b>
<b>K Tabellarischer Definitionskatalog der Kernobjekte</b>	<b>88</b>
<b>L Quellenangaben</b>	<b>105</b>

## Dokument-Metadaten

### Dokumentebene: Projekt-Ebene und Meta-Audit-Ebene

Feld	Inhalt
Dokumenttitel Codebasis Prüfstatus	Roadmap für die Migration von REAL-OQS v0.0605 nach CNNA Archiv REAL-OQS_v0.0605_full_no_comments.zip Archivgebundene Migrationsroadmap; aktuelles Artefakt als PDF kompilierbar. Die Roadmap selbst ist keine modulweise Vollbeweisdokumentation, sondern ein pfad- und handoffgebundenes Auditdokument. Auf dem kritischen Pfad gelten keine freien <code>sorry/admit</code> /Axiome als zulässige Stütze; wo private Hilfssätze, Instanz-Synthese oder <code>omit</code> -Kontexte für die spätere Vollformalisierung relevant werden, müssen sie in Kontextblatt, Handoff- oder Kettenblatt explizit sichtbar gemacht werden.
Geltungsbereich	Pfeiler A–E, aktive Migrationspfade, Gate- und Stop-Regeln, Moduldisposition, Handoffs, Kettenblätter, Audit-Anhänge; keine atomare Voll-Dokumentation jedes einzelnen Lean-Beweises des Archivs.
Zielpublikum	Reviewer ohne laufenden Codezugriff, Projektmitglieder, spätere Fachreviewer für Mathematik, AQFT, OQS, Geometrie und Matter-Pfade.
Sprachstufe Quellenbasis	Mathematik first, Erläuterung second, Interpretation third. Archivcode v0.0605; CNNA-Spezifikation v0.1; Lean-Dokumentationsregelwerk v1.8; gezielt eingesetzte Primärliteratur für AQFT/LCQFT/NCG/String/Scattering.
Audit-Konvention	Tragende Aussagen werden über Lean-Namen, Modulpfade, Handoff-Tabellen, Kettenblätter, Meta-Audit-Marker, Register- und Anhangsverweise rückverfolgbar gemacht.
Geltigkeitsbindung	Bindung an das konkrete Archiv, nicht an einen Git-Commit. Fortschreibungen müssen Delta-Änderungen an Pfaden, Gates, Handoffs, Registern und Auditpunkten kenntlich machen.
Reifegrad-Konvention	Projektintern: <code>Seed</code> , <code>Scaffold</code> , <code>Kernpfad</code> , <code>Window</code> . Regelwerksnahe Abbildung: <i>vollständig bewiesen</i> , <i>strukturell vollständig</i> / <i>Eigenschaften als Prop-Felder</i> , <i>reiner Interface-Hook</i> .
Dokumentationstyp	Vollständige Projekt-/Pfad-Roadmap mit Handoff-, Ketten- und Meta-Audit-Anhängen; keine Einzeltheorem-Dokumentation im Sinne eines vollständigen Beweisbandes.

## 1 Einleitung

### Dokumentebene: Projekt-Ebene

CNNA ist in dieser Roadmap kein bereits abgeschlossenes System, sondern ein *geplantes Projekt*: Gemeint ist der kontrollierte Übergang von der archivgebundenen REAL-OQS-Konfiguration v0.0605 zu einer neuen Stammarchitektur, in der die heute im Archiv dominierende *bright-/boundary-first* Lesart methodisch zurücktritt und durch eine *derived-only*, *sector-first* und komplementnetzbasierte Kernlogik ersetzt wird. Die Roadmap beschreibt daher nicht die Verteidigung eines fertigen Resultats, sondern den formalen Rahmen eines Umbaus, in dem ontischer Kern, AQFT-Schicht, offene Dynamik, emergente Geometrie und späte Matter-/Gauge-Faces aus einer gemeinsamen CNNA-Stammtheorie hervorgehen sollen.

**Projektstatus.** CNNA wird hier ausdrücklich als *geplantes* und noch nicht vollständig realisiertes Projekt gefasst. Die Roadmap ist deshalb weder ein Abschlussbericht noch eine bloss motivische Vision, sondern ein bindendes Zwischenartefakt zwischen vorhandenem Archiv, interner Spezifikation, Lean-Regelwerk und den späteren Implementationsentscheidungen.

Der Rahmen von CNNA ist durch drei methodische Entscheidungen bestimmt. Erstens gilt *derived-only*: tragende physikalische Aussagen sollen möglichst nicht durch freie Zusatzannahmen, sondern aus formal kontrollierten Strukturen, Handoffs und Gates folgen. Zweitens gilt *sector-first*: *Bright*, *Dark* und *Interface* werden nicht als nachträgliche Dekoration eines bereits fertigen *boundary*-Pfads behandelt, sondern als primäre Organisationsform der Architektur. Drittens gilt *stammtheoretische Mehrgesichtigkeit*: CNNA soll nicht nur einen einzigen Darstellungsstil der QFT liefern, sondern mehrere QFT-Faces — AQFT, lokal kovariante, modulare, streutheoretische, renormierte und späte Matter-/Gauge-Fenster — als regime- und sektorabhängige Ausprägungen einer gemeinsamen tieferen Struktur tragen.

Daraus folgt zugleich eine negative Abgrenzung. CNNA ist in dieser Fassung weder als fertige Physiktheorie noch als vollständig formalisierter Lean-Kern präsentiert. Die Roadmap beansprucht lediglich, dass der Übergang vom Altarchiv zu einer solchen Stammtheorie *architektonisch* planbar, auditierbar und regelwerksgebunden formuliert werden kann. Gerade deshalb spielt die Unterscheidung zwischen portierbarem Kern, neu zu schreibenden Modulen, *Bright-Recovery*, *Seeds*, *Scaffolds* und *Windows* eine zentrale Rolle: Sie markiert, was bereits als belastbarer Träger der späteren Theorie gelten darf und was vorerst nur als Anschluss- oder Spezialfallstruktur geführt wird.

Die Rolle dieser Roadmap ist damit doppelt. Einerseits ist sie ein Migrationsdokument, das den vorhandenen Archivstand in operative Arbeitspakete, Stop-Regeln und Handoff-Punkte zerlegt. Andererseits ist sie ein Projektvertrag für das geplante CNNA: Sie legt fest, in welchem Sinn die künftige Architektur als komplementnetzbasierter Stammttheorie verstanden werden soll, welche Freiheitsgrade geschlossen werden müssen, welche alten Boundary-Pfade nur regressionsweise weiterleben dürfen und unter welchen Bedingungen spätere D- und E-Schichten überhaupt als derived-only anschlussfähig gelten. In diesem Sinn ist die Einleitung nicht bloss Motivation, sondern die ausdrückliche Rahmensetzung für das Projekt CNNA als nächste Entwicklungsstufe jenseits von REAL-OQS v0.0605.

## 2 Ziel und Geltungsbereich

**Dokumentebene:** Projekt-Ebene

### 2.1 Ziel

**Dokumentebene:** Projekt-Ebene

Dieses Dokument legt eine konkrete Migrationsvorgehensweise für das neue Projekt *Complement-Netzarchitektur* (CNNA) fest. Ziel ist nicht ein kosmetischer Fork des bisherigen boundary-first Pfads, sondern die Rekonstruktion einer gemeinsamen CNNA-Stammtheorie, aus der mehrere etablierte QFT-Faces — AQFT, lokal kovariante Varianten, funktoriale/glueing-orientierte Fenster, konforme Spezialregime und späte Standardmodell-/SMEFT-Fenster — sektorabhängig emergieren. Das motivische Rückgrat ist die interne CNNA-Spezifikation: *derived-only*, *sector-first*, organisierte Komplementfamilien, eigener Interface-Träger, Parameter-Closure anstelle freier ontischer Cutoffs und explizite Bright-Recovery nur als später Spezialfall. [2, pp. 1–10, 20–36]

### 2.2 Geltungsbereich

**Dokumentebene:** Projekt-Ebene

Die Aussagen sind an das Archiv `REAL-OQS_v0.0605_full_no_comments.zip` gebunden. [3, p. 7] Die Roadmap deckt den vorhandenen Codebestand der Pfeiler A–C ab und plant darauf aufbauend die neuen Pfeiler D und E. Sie ist *keine* theorematistische Voll-Dokumentation jedes vorhandenen Lean-Beweises, sondern eine architekturgebundene Migrationsschrift mit Modul-, Herkunfts- und Gate-Bindung.

### 2.3 Was dieses Dokument nicht behauptet

**Dokumentebene:** Projekt-Ebene

- Es behauptet nicht, dass die physikalische Endtheorie bereits konstruiert ist.
- Es behauptet nicht, dass die späten Windows (Minkowski, Conformal, FLRW, SM/SMEFT) schon zum Kern gehören.
- Es behauptet nicht, dass einzelne Pfade, wie das Komplement-Netzwerk, AQFT, andere QFT Faces, Thermodynamik, Raumzeit, asymptotic safety, induzierte Gravitation oder gar das Standardmodell-Matching schon formal geschlossen vorliegen.

**Startpunkt der strikten Migration:** das aktuelle REAL-OQS-Archiv v0.0605. **Zwischenobjekte:** portierbarer Kern aus A–C, neue Core/ToC/OQS/AQFT/Integration-Schicht, anschliessend D- und E-Strang. **Endpunkt:** eine gemeinsame CNNA-Stammtheorie, in der der alte bright/boundary-first Pfad nur noch als später rekonstruierbarer Spezialfall erscheint.

**Leitsatz der späteren Symmetrie-/Materiepfade:** Die full-derived Route zu vN-/Typ-III-, CPT-, Kramers- und Spin-Statistik-Aussagen läuft nicht über Bright-Recovery, sondern über die komplementnetz-basierte modulare Darstellungsarchitektur. Das Komplementnetz ist hier kein projektinterner Stilzug, sondern der physikalische Träger dafür, dass Lokalität, Zustand, Darstellung, Modularität, Kausalität/Geometrie und spätere Symmetrie-/Statistiksätze aus demselben derived-only Pfad hervorgehen. Für belastbare Charge-/Gauge-, Nuclearity-, lokal-kovariante Dynamik- und renormierte Lokalfeld-Aussagen werden darüber hinaus explizit eine DHR/DR-artige Sektor-/Feldrekonstruktionsschicht, eine Nuclearity-Schicht, relative Cauchy-Evolution/dynamical locality sowie Hadamard-/mikrolokale und lokale Wick-/Zeitordnungs- produkte benötigt.

### 3 Dokumenttyp, Quellenhierarchie und Minimal-Audit

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Das Regelwerk fordert für vollständige oder pfadtragende Dokumente mindestens Ziel/Geltungsbereich, Dokumenttyp, Definitionskatalog, Projektglossar, globale Kontexte, Abhängigkeitsgraph bzw. Strukturkarte sowie Meta-Audit-Marker.[3, pp. 7–8] Diese Roadmap folgt derselben Logik in einer architektonischen, nicht in einer beweisdokumentierenden, Form.

#### 3.1 Quellenhierarchie

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

1. primäre technische Wahrheitsquelle: der inspectierte Lean-Archivbestand v0.0605;
2. primäre Projektsemantik für das Zielsystem aber mittlerweile unterdeterminiert: *Cnna Project Spec V0 1.pdf*;
3. normativer Dokumentationsrahmen: *lean\_doku\_regelwerk\_v1\_8.pdf*;
4. externe Physik-/Math-Physik-Literatur nur dort, wo sie die Wahl der Zielarchitektur oder die Gate-Reihenfolge rational motiviert.

#### 3.2 Reifegrad-Konvention und Abbildung auf das Regelwerk

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Die Roadmap verwendet projektintern die Klassen `Seed`, `Scaffold`, `Kernpfad` und `Window`. Für die regelwerksnahe Auditierbarkeit werden diese Begriffe wie folgt rückgebunden:

- `Seed/Window`: in der Regel *reiner Interface-Hook* oder bewusst offenes Spezialfenster;
- `Scaffold`: in der Regel *strukturell vollständig, Eigenschaften als Prop-Felder* oder noch nicht de-seedete Tragerstruktur;
- `Kernpfad` mit geschlossenen Gates: Ziel ist *vollständig bewiesen*; wo dies in der Roadmap nur projektiv geplant ist, bleibt der Status explizit als noch nicht geschlossen markiert.

Damit wird vermieden, dass projektinterne Reifegrade sprachlich stärker klingen als die regelwerksnahe Auditklassifikation.

#### 3.3 Zwang zur lesbaren Notation für alle Lean-Objekte

**Dokumentebene:** Projekt-Ebene, Pfad-Ebene und Meta-Audit-Ebene

Das Regelwerk fordert für tragende und wiederkehrende Kernaussdrücke eine explizite Lesbarkeitsnotation; für CNNA wird diese Pflicht hier bewusst verschärft: *alle Lean-Objekte auf dem aktiven Pfad müssen entweder bereits als menschenlesbare Namen vorliegen oder von einer explizit registrierten lesbaren Notations- bzw. Alias-Schicht begleitet werden*. Das betrifft nicht nur wiederkehrende Formeln, sondern ebenso Struktur- und Def-Namen, projektinterne Statusmarker, Transportabbildungen, Handoff-Objekte und Theoremgruppen, sobald ihre rohe Lean-Form die direkte mathematische Lesbarkeit verdeckt.

Die beigegefügte Notationsskizze zeigt das intendierte Muster: eine eigene Lean-Notationsdatei führt über `notation` bzw. `scopednotation` eine rein syntaktische, semantisch neutrale Lesbarkeitsschicht ein. Der Compiler sieht denselben Term wie zuvor; nur die Darstellung für Menschen wird mathematisch lesbar. Genau dieses Muster wird für CNNA zum verbindlichen Designprinzip:

- jede tragende Objektfamilie des aktiven Pfads erhält eine dokumentierte lesbare Form;

- die Notationsschicht lebt nach Möglichkeit *in Lean selbst* (z. B. `CNNA/Notation.lean`, `CNNA/PillarA/Notation.lean`, `CNNA/AQFT/Notation.lean`) und nicht bloss als externe PDF-Umschrift;
- die Notation erzeugt *keinen* neuen mathematischen Gehalt, sondern expandiert auf bereits definierte Terme, Strukturen oder Kompositionen;
- bei der ersten Verwendung im Text müssen Lean-Name, lesbare Form und Expansion bzw. semantische Bindung zusammen sichtbar sein;
- für neue Versionen größer als v0.0605 gilt: roher Name allein ist für tragende Objekte nur dann zulässig, wenn er bereits ohne Zusatz voll menschenlesbar ist.

Damit verschiebt sich die Lesbarkeitsforderung von einer optionalen Dokumenthilfe zu einer echten Architekturpflicht. Nicht nur wiederkehrende Ausdrücke wie quadratische Formen, Spuren, Adjunktionen oder Exponentialterme, sondern *alle* relevanten Lean-Objektnamen des CNNA-Kerns müssen für Reviewer in eine mathematisch oder fachlich lesbare Schicht übersetzt werden. Die Roadmap behandelt diese Schicht deshalb nicht als Kosmetik, sondern als Teil der Auditierbarkeit des Codes selbst: wer einen Namen nicht lesen kann, kann auch seine Rolle in Handoff, Kette, Gate und Beweisziel nicht verlässlich reviewen.

**Verbindliche Notationsregel.** Für den aktiven CNNA-Kern darf kein tragender Lean-Objektnamen dauerhaft nur in seiner rohen Projektschreibweise verbleiben. Entweder der Name ist bereits selbstsprechend, oder es muss eine registrierte lesbare Form existieren. Diese lesbare Form SOLL über `notation/scopednotation` in Lean realisiert werden; eine reine PDF-Umschrift ohne Lean-seitige Entsprechung genügt nicht als Zielzustand.

### 3.4 Warum eine Stammtheorie mit mehreren Faces?

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Dedushenko betont im Snowmass-White-Paper, dass keine der heute verfügbaren Axiomatik-Familien alle physikalischen Erscheinungsformen von QFT abdeckt. Die Übersicht behandelt gerade deshalb nebeneinander Wightman, OS, AQFT/Haag-Kastler, perturbatives AQFT, lokal kovariante QFT, homotopisches AQFT, Functorial Field Theory und CFT/conformal nets.[4, pp. 1, 4–12] Genau daraus folgt für CNNA eine scharfe Designkonsequenz: das Ziel darf nicht sein, eine einzige übernommene Oberflächenbeschreibung zu verabsolutieren; das Ziel muss eine tieferliegende, regime-sensitive Stammstruktur sein, aus der mehrere Faces konsistent hervorgehen.

### 3.5 Aktuelle Größenordnung des inspectierten Archivs

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Die Codebasis v0.0605 ist in ihrer heutigen Form noch asymmetrisch: A und B tragen den Grossteil des mathematischen Gewichts, während C kompakt ist und D/E als neue CNNA-Struktur noch fehlen.

**Tabelle 2: Aktuelle Größenordnung des inspectierten Archivs – Teil 1**

Pfeiler	Lean-Dateien	LOC
Pillar A	127	11'073
Pillar B	89	12'412
Pillar C	18	648

Die reine Pfeilerstatistik ist aber nicht die ganze operative Last. Im Archiv existieren zusätzlich Root-, Meta- und Example-Dateien, die zwar nicht als fachlicher Stamm fungieren, aber Importstabilität, Build-Gates und Regressionsfähigkeit tragen. Sie müssen deshalb in der Migration explizit mitgeplant werden und dürfen nicht zwischen Pillar und “blosser Infrastruktur” verschwinden.

Tabelle 4: Aktuelle Größenordnung des inspectierten Archivs – Teil 2

Block ausserhalb der Pillar-Statistik	Lean-Dateien	LOC
Top-Level-Infrastruktur (Basic, BuildAll, PillarA/B/C)	5	8
Meta	19	908
Examples	14	1'026
Summe ausserhalb A–C	38	1'942

Diese Asymmetrie ist inhaltlich wichtig: die Migration darf D und E nicht durch freie Platzhalter simulieren, sondern muss sie aus dem in A–C bereits vorhandenen Material ableiten. Zugleich ist damit klargestellt, dass *auch* die ausserhalb der Pillar-Tabelle liegenden Dateien echte Migrationslast tragen: Meta-Dateien binden Regression und Audit, Examples sichern Replay-Pfade, und die Root-Dateien tragen Toolchain-, Import- und Build-Stabilität.

### 3.6 Roadmap als lebendes Gate-Dokument: prototypisches Vorseilen und Rückfluss von Erkenntnissen

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Diese Roadmap ist bewusst kein starrer Master-Phasenplan, sondern ein lebendes Gate-Dokument. Sie wird ab jetzt ausdrücklich wie ein *executable Dokument* behandelt: jede neue belastbare Erkenntnis aus dem aktiven Lean-Codepfad erzwingt ein Update der Roadmap, und jedes belastbare Roadmap-Update erzeugt seinerseits über Tabellen, Register, Handoffs und Kettenblätter die neue Ausgangsbasis für Notationsschicht, Audit-Anhang und gegebenenfalls auch für die partielle Ordnung selbst. Die Roadmap lebt damit nicht nur entlang des Vorwärtspfads, sondern *nach allen Seiten*: in Richtung Code, Dokumentation, Notation, Gate-Reife und Rückfluss auf frühere Architekturentscheide. Gerade in Lean führen zu früh verhärtete Begriffs- und Typentscheidungen leicht in eine *Typ-Sackgasse*: eine in A scheinbar harmlose Festlegung kann später B-, C- oder D-Pfade unverhältnismäßig teuer machen, wenn sie die benötigten Darstellungs-, Restriktions- oder Glueing-Schichten nicht mehr trägt. Deshalb wird detaillierte Nachfolgeplanung erst dort hart, wo der vorgelagerte Pfeiler sein Gate wirklich geschlossen hat; vorher bleibt sie begründete, aber bewusst revidierbare Absicht.

Zugleich erlaubt die Roadmap ein *prototypisches Vorseilen*: in A dürfen früh skizzierte Dummy- bzw. Sketch-Schnittstellen für spätere Verbraucher aus B/C/D angelegt werden, etwa für `ComplementLocalNet`-nahe Restriktionen, Zustandsprojektionen, modulare Aktionen oder Glueing-Signaturen. Solche Vorgriffe zählen ausdrücklich nicht als fertiger Kern, sondern müssen als **Seed** oder **Scaffold** markiert bleiben. Ihr Zweck ist nicht, spätere Mathematik vorzutauschen, sondern früh zu testen, ob die Typ- und Exportlage von A die Last der späteren Schichten überhaupt tragen kann.

Weil CNNA als Stammtheorie auf ungeplante Emergenz und Strukturfeedback reagieren muss, fungiert die Roadmap zusätzlich als *Logbuch von Architekturentscheidungen aus Notwendigkeit*. Wenn im aktiven Codepfad neue Symmetrien, Selektoren oder Restriktionsnotwendigkeiten sichtbar werden, müssen die Anforderungen an B/C/D sofort nachgeführt werden; umgekehrt darf eine spätere Einsicht aus C, D oder E frühere A-Entscheidungen wieder öffnen, falls sich deren Schnittstellen als zu restriktiv erweisen.

Daraus folgt eine explizite *Interface-Stabilitätsregel*: Erkenntnisrückfluss nach früheren Pfeilern ist zulässig, aber nur kontrolliert. Sobald ein späterer Pfeiler zeigt, dass ein früheres Axiom, eine Typwahl oder ein Exportvertrag zu eng war, wird das betroffene Modul zunächst wieder auf **Seed/Scaffold**-Status zurückgestuft, die Source-Maps und Imports werden angepasst, Regression und Replay werden erneut geprüft, und erst danach darf das Gate wieder geschlossen werden. Genau dafür existiert die Gate-Logik: sie garantiert Flexibilität *vor* Closure und Härte *nach* Closure, statt einen unfertigen Unterbau durch parallele Detailplanung zu überfahren.

### 3.7 Mehrstufiges Wahrheitsprotokoll: AI, Formalisierung, Verifikation, Semantik und Physik

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

CNNA benutzt AI nicht als fachliche Autorität, sondern als heuristische Vorschlags-, Strukturierungs- und Übersetzungsschicht zwischen informeller Mathematik, formaler Sprache, maschinell geprüfem Code und später physikalischer Deutung. Wahrheitsquelle auf dem kritischen Pfad ist deshalb nicht eine plausible AI-Prosa, sondern die Kombination aus wohldefinierten Definitionen, strikt typgeprüfem Lean-Code, Kernel-Check und einer expliziten Disziplin gegen `sorry`, `admit` und freie Axiome im aktiven Kernpfad.

Damit verschiebt sich der methodische Engpass. Nicht das freie Formulieren eines scheinbar eleganten Beweises ist das Primärproblem, sondern die Wahl tragfähiger Definitionen, Typ- und Interface-Entscheidungen, sauberer Handoff-Strukturen sowie die *korrekte Prosa-Semantik* der Kommentare, Dokumente und Brückentexte. Gerade der letzte Punkt ist nicht kosmetisch: der Code kann formell korrekt sein, während seine menschliche Beschreibung falsch, zu stark oder physikalisch irreführend ist. Deshalb gehören Kommentar- und Dokumentations-Cleanup, semantische Gegenprüfung und explizite Herkunftsbindung selbst zur wissenschaftlichen Absicherung. Gerade an allen kritischen Übergängen — Definition, Handoff, Kommentarsemantik, externe Dokumentation und physikalische Deutung — bleiben Reviews durch fachlich qualifizierte Expertinnen, Experten und professionelle Reviewer unverzichtbar.

Methodisch ergibt sich daraus ein mehrstufiges Wahrheitsprotokoll:

1. **Vermutung / informelle Idee:** eine mathematische oder physikalische Intuition wird als Arbeitsvermutung formuliert;
2. **AI-gestützte Strukturierung:** AI darf Begriffe sortieren, Ableitungsketten skizzieren, Literaturpfade verdichten und Übersetzungsarbeit zwischen informeller und formaler Sprache leisten;
3. **Formalisierung:** die Aussage wird in Lean-Definitionen, Interfaces, Handoffs und saubere Theoreme überführt;
4. **Maschinelle Verifikation:** wahr im technischen Sinn ist nur, was der Proof-Checker relativ zu den gewählten Definitionen wirklich akzeptiert;
5. **Semantikprüfung und Fachreview:** Menschen sowie fachlich qualifizierte Expertinnen, Experten und professionelle Reviewer prüfen, ob Kommentare, externe Dokumente, Source-Maps und Theorembeschreibungen den kompilierten Code korrekt wiedergeben;
6. **Physikalische Interpretation:** erst danach wird durch fachlich qualifizierte Reviews bewertet, ob der formale Satz für die intendierte Physik relevant, stark genug und nicht fehlgedeutet ist.

Dieses Protokoll ist bewusst strenger als die übliche Redeweise “AI hat bewiesen, dass .... Im CNNA-Kontext gilt stattdessen: *AI darf vorschlagen; wahr und belastbar ist nur, was formal trägt, semantisch korrekt dokumentiert ist und die physikalische Deutung übersteht.* Gerade für derived-only-Architekturen ist diese Trennung zentral, weil ein Proof-Checker nur relativ zu den gewählten Definitionen prüft, nicht aber automatisch deren physikalische Relevanz, Nicht-Trivialität oder Emergenzstärke.

**Methodische Leitregel:** AI ist in dieser Roadmap weder mathematische noch physikalische Letztautorität. Sie dient als Heuristik- und Übersetzungsschicht. Autoritativ sind auf dem aktiven Pfad nur (i) die formale Sprache und der Kernel des Proof Checkers, (ii) die expliziten Projektregeln gegen `sorry/admit`/freie Axiome, (iii) die semantische Gegenprüfung von Prosa und Kommentaren und (iv) die nachgelagerte physikalische Interpretation. Damit ist die Roadmap nicht bloss “AI + Lean, sondern ein kontrolliertes mehrstufiges Wahrheitsprotokoll. In allen kritischen Instanzen bleiben Reviews durch fachlich qualifizierte Expertinnen, Experten und professionelle Reviewer unabdingbar; weder AI-Prosa noch blosses Typchecking ersetzen diesen Schritt.

### 3.8 Operative Verdichtung des aktiven Migrationspfads

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Die bisherige Struktur lässt sich für den aktiven Projektvollzug auf eine knappe Kernlogik verdichten. Erstens bleibt diese Schrift eine *architekturgebundene Migrationsroadmap* für das Archiv `REAL-OQS_v0.0605_full_no_comments.zip`; sie ist keine lose Wunschliste und keine ex post erzielte Ergebnissnarration. Zweitens ist das Ziel eine *CNNA-Stammtheorie* im strict-derived-only- und sector-first-Sinn, aus der mehrere QFT-Faces emergieren; der alte Bright-/Boundary-Pfad ist dafür später Spezialfall, Recovery und Regressionsträger, aber nicht mehr begriffliches Zentrum. Drittens trägt die Architektur genau die Pfeilerfolge  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ : ontischer Kern und Komplementgrammatik, AQFT aus Komplementnetz, offene Systeme/Kanäle, emergente Geometrie/LocalCovariance und erst danach späte Gauge-/Matter-/SM-Fenster.

Gerade deshalb müssen die Stop- und Gate-Bedingungen im Vollzug sichtbar bleiben. Auf dem aktiven Pfad darf `ComplementSpectrumCondition` nicht hinter `BisognanoWichmannSeed` verschwinden; die modulare B-Kette `ComplementGNS`  $\rightarrow$  `SeparatingProperty`  $\rightarrow$  `TomitaTakesakiData`  $\rightarrow$  `ModularConjugation`  $\rightarrow$  `ComplementKMS` ist keine stilistische Ausdifferenzierung, sondern die harte Vorbedingung späterer  $vN$ -/Typ-III-, CPT-, Kramers- und Spin-Statistik-Lesarten. Ebenso ist `ParameterClosure` keine späte Schönheitskorrektur: ohne `selectedBranching`, `HorizonLevel`, `BackreactionFixedPoint` und `RegularizationClosure` bleibt `derived-only` unvollständig, weil ontische, dynamische oder numerische Restparameter sonst unkontrolliert aus dem Kernpfad herausragen.

Daraus folgen für den unmittelbaren Arbeitsvollzug fünf Prioritätsregeln:

1. **A-Core zürst einfrieren.** `Approximant`, `RegionCore`, `RegionNet`, `InfiniteCarrier`, `Selection`, dazu `BranchPatch/ComplementSectorFamily/SectorSplit`, werden semantisch stabilisiert, bevor tiefere Recovery-Pfade nachgezogen werden.
2. **GeneralizedDtN als explizite Kernstruktur bauen.** Die Mehrsektor-Schur-/DtN-Schicht bleibt ein `structure`-basiertes Kernobjekt mit klaren Restriktions-, Kompositions- und Glueing-Daten; die Verbindung zu `ComplementLocalNet` wird früh getestet, nicht erst am Ende.
3. **Komplement- und Interface-Netze vor Bright-Recovery.** `ComplementLocalNet` und `InterfaceLocalNet` entstehen vor jeder portierten `BoundaryMatrix-Comfort`-Schicht. Alte `BoundaryMatrix*`-Dateien zählen nur als Replay-, Regression- und Extraktionsquelle und sollten organisatorisch als *bright-recovery*-Unterstrang behandelt werden.
4. **Parameter-Closure früh aktivieren.** Der operative Closure-Pfad `BranchingWitness`  $\rightarrow$  `selectedBranching`  $\rightarrow$  `HorizonLevel/BackreactionFixedPoint`  $\rightarrow$  `RegularizationClosure` läuft früh mit, damit  $b$ ,  $L_{\max}$  und Regularisierungsreste nicht als stille Aussenparameter stehen bleiben.
5. **Namespace- und Semantik-Sauberkeit erzwingen.** Portierte Module dürfen keine implizite `boundary-first`-Semantik in Bezeichnern, Kommentaren, Source-Maps oder Imports

weitertragen; wo das Altarchiv nur Bright-Spezialfall kodiert, muss dies sichtbar gemacht und aus dem CNNA-Kern herausgezogen werden.

**Operatives Risikobild:** Die kritischen Migrationsrisiken liegen nicht mehr in fehlenden grossen Pfeilern, sondern in stillen Altsemantiken: trunkartige BoundaryMatrix-Imports im Kernpfad, freie Restparameter  $b$ ,  $L_{\max}$ ,  $\beta$  ohne Closure, sowie Namens- und Kommentarspuren, die portierte Module weiter als boundary-first lesen lassen. Die Gegenmassnahme ist nicht mehr Prosa, sondern frühes Einfrieren des A-Kerns, aktive Closure, harte Gate-Disziplin und konsequente Trennung zwischen CNNA-Kern und Bright-Recovery.

## 4 Definitionskatalog der Kernobjekte

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

Dieser Abschnitt trägt die formale Last der im aktiven und spät aktivierten Pfad wiederkehrenden Kernobjekte. Er ist kein erläuterndes Glossar, sondern die *kanonische Objektquelle* der Roadmap. Jede Zeile definiert genau *ein* kanonisches Lean-Objekt. Register und Notations-Appendix dürfen keine neuen Fachobjekte erzeugen, sondern müssen auf diesen Katalog zurückverweisen.

### 4.1 Formaler Definitionskatalog des aktiven und spät aktivierten Pfads

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

Die folgende Liste führt nicht nur Name, Schreibweise und Rolle, sondern für jedes Objekt auch Objektklasse, primäre lesbare Form, Lesbarkeitsmodus, exakte Bindung, Zielmodul der möglichen Notations-Emission, Scope und Pfadstatus. Damit ist sie die kanonische Eingabemenge für Register, Notations-Appendix und spätere `Notation.lean`-Module.

### 4.2 Notationspflicht innerhalb des Definitionskatalogs

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Für jeden vollen Kerneintrag gelten ab jetzt die folgenden Integritätsregeln:

1. **Eine Zeile = ein kanonisches Objekt.** Zusammgedrückte Sammelzeilen sind für ableitungsrelevante Objekte unzulässig.
2. **Primäre lesbare Form.** Jeder Eintrag trägt genau eine kanonische Oberflächenform; weitere Aliasformen gehören ins Register.
3. **Exakte Expansion / Bindung.** Jede lesbare Form wird explizit auf den zugrunde liegenden Lean-Namen zurückgeführt.
4. **Expliziter Lesbarkeitsmodus.** Zulassungen für *selfspeaking*, *text alias*, *symbolic notation* und *symbolic notation + text alias* werden pro Objekt festgelegt.
5. **Keine technischen Alternativen in einer Zelle.** Formulierungen wie *oder*, *bzw.* oder *später optional* sind in der ableitungsrelevanten Tabellenlogik verboten.
6. **Notations-Emission nur mit Zielmodul und Scope.** Jede exportierte Lesbarkeitsform benötigt eine eindeutige Modul- und Scope-Bindung.

Meta-Begriffe und Reifegradmarker werden *nicht* in diesen Definitionskatalog eingemischt, sondern später in eigene Registertabellen ausgelagert.

Für den Definitionskatalog gelten folgende Integritätsregeln:

1. Jede aktive oder spät aktivierte tragende Code-Entität erscheint genau einmal.
2. Jede Zeile besitzt genau eine Primärform; Exportformen sind strikt sekundär.
3. Jeder Eintrag mit Modus ungleich *selfspeaking* oder *no exported readability layer* erzeugt mindestens einen Registereintrag.
4. Jeder Eintrag mit exportierter Form erzeugt mindestens einen Appendix-Eintrag.
5. Namespace, Aktivierungsregel, Kontext, Kollisionspolitik, logische Vorgänger und Verbraucher/Nachfolger dürfen nicht ausschliesslich proseartig ausserhalb der Tabelle festgelegt werden.

**Tabellarische Auslagerung.** Der vollständige tabellarische Definitionskatalog ist aus Layoutgründen in den Anhang ausgelagert; vgl. Anhang K. Im Haupttext verbleiben nur die Einordnung des Katalogs und die normativen Integritätsregeln seiner Verwendung.

## 5 Projektglossar und Verweißstruktur

Dokumentebene: Projekt-Ebene

### 5.1 Kernbegriffe des Projekts

Dokumentebene: Projekt-Ebene

Tabelle 6: Kernbegriffe des Projekts – Teil 1

Begriff	Rolle in der Roadmap
ToC	kanonisches ontisches Substrat; in CNNA nicht mehr nur Quelle eines Bright-Randnetzes, sondern Generator von Bright-, Dark- und Interface-Struktur. Mathematisch braucht der aktive Pfad mindestens ein verzweigungsfähiges, geankertes Cliques-/Pfadobjekt mit Inzidenz-, Koarsenierungs-, Branching- und Source-Map-Daten, so dass BranchPatch, ComplementSectorFamily und SectorSplit nicht frei schweben. Diskrete, kontinuierliche oder graduierte ToC-Varianten sind nur dann zulässig, wenn sie genau diesen sektor-split-kompatiblen Vertrag instanziiieren, statt eine neue Ontologie an der Split-Schicht vorbei einzuschmuggeln.
Bright	expliziter endlicher REAL-Approximant bzw. sichtbarer Patch; wichtig, aber nicht mehr privilegiertes Endprodukt.
Dark	organisierte Komplementfamilie inklusive Trunk, Seitenästen und eigenen UV-Tails; keine blöÙe Restumgebung.
Interface	eigener mathematischer Träger für Kopplung, Rückwirkung, Entropie, Modularität, Glueing und spätere Defekt-/Dualitätsdaten.
BranchPatch	erster endlicher sichtbarer Patch relativ zu einem ToC-Anker; ersetzt die alte implizite Zentrierung auf eine feste Randoberfläche.
ComplementSectorFamily	organisierte Familie komplementärer Sektoren; ersetzt das alte singuläre Komplement.
SectorSplit	expliziter Mehrsektor-Split, der Bright, Dark und Interface synchronisiert.
GeneralizedDtN	verallgemeinerter DtN-/Schur-Kopplungskern für mehrere Komplementsektoren; algebraisch keine bloÙe Zahl, sondern sektorindexierte block- bzw. operatorwertige Kopplungsstruktur mit Restriktions-, Kompositions-, Symmetrie-/Positivitäts- und Glueing-Daten. Der alte binäre DtN-Fall bleibt nur Sonderfall. Lean-seitig sollte diese Schicht primär als structure mit expliziten Feldern und nachgelagerten Instanzen/Gates für Restriktion, Lokalisierung und Verträglichkeit mit ComplementLocalNet modelliert werden, nicht als nackte Typeclass oder lose Kategoriefassade.
TruthProtocol	mehrstufiges Wahrheitsprotokoll für CNNA: Vermutung, AI-gestützte Strukturierung, Formalisierung, maschinelle Verifikation, Semantikprüfung und physikalische Interpretation bleiben explizit getrennt; technische Wahrheit entsteht nicht durch AI-Prosa, sondern erst nach Kernel-Check und semantischer Gegenprüfung.
SemanticReview	verpflichtende Kontrolle, ob Kommentare, externe Dokumente, Source-Maps und Theorembeschreibungen den kompilierten Lean-Code korrekt wiedergeben; verhindert, dass formell korrekter Code durch falsche Prosa fehlgedeutet wird.
ExpertReview	unabdingbare fachliche Gegenprüfung durch qualifizierte Expertinnen, Experten und professionelle Reviewer; prüft Definitionen, Relevanz, Semantik, physikalische Deutung und die Belastbarkeit des formalen Pfads über das reine Typchecking hinaus.
ComplementLocalNet	lokales Netz auf dem dunklen bzw. komplementären Sektor.
InterfaceLocalNet	eigenes lokales Netz auf dem Interface; gerade hier sitzt die spätere Glueing-, Entropie- und Defektstruktur.
ComplementStateNet	Zustandsfamilien auf Bright, Dark und Interface; keine freie State-Signatur auf dem kritischen Pfad.
ComplementGNS / ComplementKMS	Repräsentations- und Thermikschicht des komplementären Netzes; bauen auf der generischen GNS/KMS-Basis auf, dürfen aber nicht ohne Separations- und Modularitätszwischenschicht identifiziert werden.
GradedStarAlgebra	Z2-offene algebraische Grundschrift; macht fermionische/statistische Sprache zu einer Rückbindung an die AQFT-Basis statt zu freier später Etikettierung.
SeparatingProperty	Reeh-Schlieder-artige Separationsschicht; stärkt die GNS-Lage von bloss zyklisch zu zyklisch+separierend für relevante Komplementalgebren.
TomitaTakesakiData	extrahiert Tomita-Operator, modulare Konjugation $J$ und modularen Operator $\Delta$ aus der komplementnetz-basierten Darstellungslage.
ModularConjugation	kontrolliert die Bright/Dark/Interface-Lesart von $JMJ = M'$ ; eigentliche modulare Darstellungsrchitektur zwischen GNS und KMS.
ComplementSpectrum	
Condition	explizite Spektral- und Vorwärtskegelkontrolle des aktiven Komplementnetzes; Vorbedingung für BW-, Guido-Longo- und spätere Scattering-Lesarten.
Bisognano	
WichmannSeed	explizite $B \rightarrow D$ -Brücke; vergleicht modulare Automorphismen mit abgeleiteten Boost-/Wedge-Strukturen und bindet Modularität an Geometrie.
NuclearitySeed	Phase-Space-/Energielevel-Dichteschicht jenseits blossen Split-Properties; beginnt erst nach der geschlossenen KMS-/Spektralkontrollschicht und stabilisiert spätere Typ-III-/Thermik-Lesarten.
DHRsectorsSeed	observablenseitige Superselection-Schicht für lokalisierte Ladungssektoren; Brücke von AQFT-Netzen zu Charge- und Statistiksprache.
FieldAlgebra	
Reconstruction	
Seed	DR-artige Rekonstruktionsstufe von Feldalgebra und kompakter Eichgruppe aus der observablen Sektorstruktur.
RelativeCauchy	
Evolution	
Seed	LCQFT-Dynamik unter Hintergrundvariation; verknüpft lokale Kovarianz mit abgeleiteter Dynamik und Stress-Tensor-Lesart.
HadamardStateSeed	mikrolokal kontrollierte Zustandsklasse für gekrümmte emergente Hintergründe; Vorbedingung für renormierte lokale Observablen.
LocalWick	
Polynomials	
Seed	lokale kovariante Wick- und Zeitordnungsprodukte; trägt spätere Stress-Tensor-, OPE- und Renormierungslesarten.
HorizonLevel	abgeleitete Horizontgröße; ersetzt ein primitives $L_{\max}$ auf dem aktiven Pfad.
ParameterClosure	pfilerübergreifender Closure-Strang für ontische Mikrostrukturparameter, dynamische Emergenzskalen und sekundäre Regularisierer; markiert die derived-only-Schwelle gegen freie Restparameter.
BackreactionFixedPoint	Fixpunktstruktur, in der Einfluss-, Entropie-, Skalen- und Zustandsdaten konsistent auf eine effektive IR-/Horizontgröße zurückwirken.
RegularizationClosure	schliesst numerische Stabilisierung entweder durch intrinsische Positivität/Koerzivitaaet oder degradiert Restterme zu abgeleiteten Spektral-/Nullmodenparametern.
EffectiveLambda	abgeleitete kosmologische Skalargröße aus Interface-Balance, Entropie oder Fixpunktdiagnostik.
SpectralDimensionFlow	abgeleitete Spektral-/Dimensionsdiagnostik; im Kern erst nach Closure.
InfiniteCarrier / quasilokaler Limes	markiert den Übergang vom endlichen Bright-Approximanten zur unendlichen/quasilokalen AQFT-Lage; Vorbedingung für belastbare Typ-III-, Tomita-Takesaki-, BW- und Scattering-Lesarten.

Begriff	Rolle in der Roadmap
PositiveEnergy / UnitaryDynamics	explizite Positivitäts- und Unitaritätskontrolle der effektiven Generator- und Entwicklungsschicht; dürfen im full-derived Pfad nicht nur implizit mitlaufen.
PrototypeInterface	bewusst vorgezogene Dummy-/Sketch-Schnittstelle für spätere Verbraucher; prüft früh, ob A-seitige Typ- und Exportentscheide B/C/D überhaupt tragen, zählt aber nie als geschlossener Kern.
InterfaceStability	Regel, dass spätere Einsichten frühere Export- oder Typverträge kontrolliert wieder öffnen dürfen; verlangt Rückstufung auf Seed/Scaffold, Regressionsprüfung und erneutes Gate-Closing.
BackPropagationOfInsights	disziplinierter Rückfluss später Architekturzwänge in frühere Pfeiler; macht die Roadmap zum Logbuch notwendiger Designentscheidungen statt zu einer starren linearen Wunschliste.
Seed / Scaffold	bewusst noch nicht de-seedete oder generische Modulhüllen; dürfen den kritischen Pfad nicht als fertige Kernbegriffe dominieren.
Window	spätes Spezialfenster (z.B. Minkowski-, Conformal-, FLRW-, SM- oder SMEFT-Fenster), das standardmäßig <i>nicht</i> zum Kern de-seedet wird.

## 5.2 Strikte Strukturkarte

### Dokumentebene: Pfad-Ebene

**Symbole und Ausschlüsse vor der Kette.** Startpunkt ist der ToC-Vertrag samt Branch Patch, ComplementSectorFamily, SectorSplit und GeneralizedDtN. Historische Bright-only-Recovery-Pfade, bloss rhetorische Matter-Fenster und ungeschlossene Closure-/Window-Sprachen sind *nicht* Gegenstand der folgenden strikten Kette.

Die alte strikte Ableitung läuft im aktiven Bright-Pfad schematisch als

$$\text{ToC} \rightarrow (\Omega, L, \beta) \rightarrow \text{BoundaryMatrixNet} \rightarrow \text{State} \rightarrow \text{GNS} \rightarrow \text{KMS} \rightarrow \text{Modular}.$$

Die CNNA-Stammstruktur muss demgegenüber lauten:

**Tabelle 8: Strikte Strukturkarte – Teil 1**

- ToC → BranchPatch / ComplementSectorFamily / SectorSplit
- GeneralizedDtN / MultiSchur
- Complement- und Interface-Netze / InfiniteCarrier / quasilokaler Limes
- Zustände / ComplementGNS / SeparatingProperty / TomitaTakesakiData
- ModularConjugation / ComplementKMS / ComplementSpectrumCondition
- Bisognano-Wichmann-Seed / Nuclearity
- Kanäle / Backreaction
- Parameter-Closure / Branching- und Regularization-Closure
- BackreactionFixedPoint / HorizonLevel / EffectiveLambda
- emergente Geometrie / LocalCovariance / RelativeCauchyEvolution
- Hadamard-/mikrolokale Zustandskontrolle / lokale Wick-Produkte
- DHR-Sektoren / Feldalgebra- und Gaugegruppen-Rekonstruktion
- CPT/Spin/Statistik-Brücke / Gauge-Matter- und SM/SMEFT-Fenster.

Die starke Lesart dieser Strukturkarte ist bewusst nicht optional: vN-/Typ-III-, CPT-, Kramers- und Spin-Statistik-Aussagen sollen im CNNA-Kern gerade *nicht* aus Bright-Recovery stammen, sondern aus der komplementnetz-basierten modularen Darstellungsarchitektur, die zwischen ComplementGNS und ComplementKMS explizit geschlossen wird. Für isolierte toy-level- Sätze in endlicher Quantenmechanik wäre eine solche Komplementarchitektur nicht in jedem Fall zwingend; für *full-derived emergente lokale QFT* ist sie es im angestrebten physikalischen Sinn jedoch weitgehend. Ohne Komplement-/Interface-Struktur entstünde höchstens eine Bright-only-Recovery auf endlichen Typ-I-Testträgern; gerade der Pfad zu modularer Konjugation, Typ-III-Lesart, Bisognano–Wichmann, CPT, Spin-Statistik und Kramers bliebe dann unvollständig oder nur aufgesetzt. Der alte Pfad wird also nicht gelöscht, aber architektonisch entzentriert: er ist Ziel eines späten Recovery- und Regressionstrangs, nicht mehr die Lesart des gesamten Projekts.[2, pp. 1–10, 30–36][42]

## 6 Pillar-Handoff-Spezifikationen

### Dokumentenebene: Handoff-Ebene

Die aktiven Pfeilerübergänge werden nicht nur rhetorisch, sondern als formale Handoff-Tabellen sichtbar gemacht. Damit bleibt prüfbar, welche Daten, Eigenschaften und Nicht-Übergaben den Paradigmenwechsel zwischen den Pfeilern tragen.

### 6.1 Handoff A → B

#### Dokumentenebene: Handoff-Ebene

Tabelle 9: Handoff A to B – Teil 1

Feld	Inhalt
Quell-Pillar / -bereich	Pillar A: ToC-Vertrag, BranchPatch, ComplementSectorFamily, SectorSplit, GeneralizedDtN, Infinite Carrier
Ziel-Pillar / -bereich Übergebene Daten	Pillar B: AQFT auf ComplementLocalNet, InterfaceLocalNet, ComplementStateNet, ComplementGNS sektorierte Regionen, Komplementfamilien, Multi-Schur-/DtN-Kopplung, Glueing-/Restriktionssignaturen, Infinite-Carrier-/Limes-Lage
Mitgegebene Beweise / Instanzen	Symmetrie-/Positivitäts- oder Restriktionsdaten nur, soweit sie in A bereits explizit hergeleitet sind; keine freie AQFT-Semantik
Nicht übergebene Struktur Rechtsstatus des Handoffs Audit-Punkte	GNS, Separierung, Tomita-Takesaki, KMS, Spektralbedingung, Haag-Kastler-/DHR-Lesart definitorischer Export plus Adapter-/Repackaging-Schicht ToC-Vertrag, GeneralizedDtN-/MultiSchur-Pfad, Source-Maps von A in die B-Netze

### 6.2 Handoff B → C

#### Dokumentenebene: Handoff-Ebene

Tabelle 11: Handoff B to C – Teil 1

Feld	Inhalt
Quell-Pillar / -bereich	Pillar B: lokale Netze, Zustände, GNS, KMS, Spektral- und Modularitätsdaten
Ziel-Pillar / -bereich Übergebene Daten	Pillar C: Kanäle, relative Entropie, Rückwirkung, offene-System-Dynamik Algebren, Zustandslagen, relevante Generatoren, Interface-Netze, thermische und spektrale Diagnostik
Mitgegebene Beweise / Instanzen	Isotonie, Lokalität, Split-/Nuclearity-Vorbedingungen, positive Energie/Unitarität soweit vorhanden
Nicht übergebene Struktur	keine freie Interpretation der Kanäle als Physik ohne explizite Rückwirkung und Fixpunktkontrolle
Rechtsstatus des Handoffs Audit-Punkte	gemischter Status: Export plus neue dynamische Konstruktion ComplementStateNet, ComplementGNS/KMS, OQS-Kanalmodule, relative Entropie und Backreaction-Source-Maps

### 6.3 Handoff C → D

#### Dokumentenebene: Handoff-Ebene

Tabelle 13: Handoff C to D – Teil 1

Feld	Inhalt
Quell-Pillar / -bereich	Pillar C: Kanäle, Rückwirkung, Entropie- und Einflussdaten, Parameter-Closure
Ziel-Pillar / -bereich Übergebene Daten	Pillar D: emergente Geometrie, LocalCovariance, Relative Cauchy Evolution, Gravity-Routen BackreactionFixedPoint, HorizonLevel, EffectiveLambda, Glueing-/Defektdaten, ggf. Spektral-diagnostik
Mitgegebene Beweise / Instanzen	Closure-Gates, positive Energie/UnitaryDynamics soweit für D nötig, Interface-Kohärenz
Nicht übergebene Struktur	keine freie Hintergrundgeometrie; keine vorab gesetzte Gravitationstheorie
Rechtsstatus des Handoffs Audit-Punkte	konstruktiver Übergang mit expliziter Nicht-Übergabe von Geometriepostulaten Parameter-Closure-Kettenblatt, RelativeEntropy-/Backreaction-Module, D-Startgates

### 6.4 Handoff D → E

#### Dokumentenebene: Handoff-Ebene

Tabelle 15: Handoff D to E – Teil 1

Feld	Inhalt
Quell-Pillar / -bereich	Pillar D: InterfaceCausality, LocalCovariance, Hadamard/Wick, Jacobson-/NCG-/induced-gravity-Routen
Ziel-Pillar / -bereich	Pillar E: CPT/Spin/Statistik, Charge/Gauge, chiral matter, Higgs/Yukawa/Flavor, SM/SMEFT-Fenster
Übergebene Daten	geometrisch-dynamische Kernlage, modulare/geometrische Brückenobjekte, Hadamard-/Wick-Kontrolle, DHR/DR-nahe Voraussetzungen
Mitgegebene Beweise / Instanzen	D-Zulässigkeit, Kausalität, renormierte lokale Observablen, ggf. spektral-geometrische Seeds
Nicht übergebene Struktur	keine freie Matter-Sprache ohne DHR/DR-, CPT- und Statistikgates; keine ersatzlose Gleichsetzung von NCG mit E-Kern

---

Rechtsstatus des Handoffs	Gate-gebundene Brücke, nicht bloss rhetorischer Themenwechsel
Audit-Punkte	D/E-Gate-Reihenfolge, Feynman-Audit, CPT-/Statistics-/Kramers-Gates, AnomalyInflow- Ordnung

---

## 7 Migrationsgesetz: was direkt portiert, was umgeschrieben und was dezentriert wird

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

### 7.1 Direkt oder nahezu direkt portierbarer Kern

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

Die folgenden Module oder Modulgruppen besitzen in v0.0605 bereits eine hinreichend allgemeine mathematische Form, um als CNNA-Unterlage zu dienen. Portierung bedeutet hier: Imports bereinigen, Namespace umstellen, Namenspolitik anpassen, aber keinen unnötigen semantischen Bruch erzeugen.

Tabelle 17: Direkt oder nahezu direkt portierbarer Kern – Teil 1

Altmodul	Ziel in CNNA	Disposition
REALOQS/PillarA/Core/Approximant.lean	CNNA/Core/Approximant.lean	nahezu direkt portieren; Namespace und Importkegel bereinigen
REALOQS/PillarA/Core/RegionCore.lean	CNNA/Core/RegionCore.lean	nahezu direkt portieren; nur Semantik auf Sektorbild umstellen
REALOQS/PillarA/Core/RegionNet.lean	CNNA/Core/RegionNet.lean	als Trager für Bright/Dark/Interface-Regionen erhalten
REALOQS/PillarA/Core/InfiniteCarrier.lean	CNNA/Core/InfiniteCarrier.lean	beibehalten; keine boundary-first Zusatzsemantik
REALOQS/PillarA/Core/Selection.lean	CNNA/Core/Selection.lean	beibehalten; Beobachterbegriff nicht frei einführen
REALOQS/PillarA/OQS/DtN.lean	CNNA/OQS/DtN.lean	als binärer Sonderfall erhalten; Ausgangspunkt für GeneralizedDtN
REALOQS/PillarA/OQS/DtNStabilized.lean	CNNA/OQS/DtNStabilized.lean	als technische Hilfsschicht erhalten, aber methodisch dezentralisieren
REALOQS/PillarB/AQFT/StarAlgebra.lean	CNNA/AQFT/StarAlgebra.lean	direkt portieren
REALOQS/PillarB/AQFT/State.lean	CNNA/AQFT/State.lean	direkt portieren
REALOQS/PillarB/AQFT/StateNet.lean	CNNA/AQFT/StateNet.lean	direkt portieren und später sektorisieren
REALOQS/PillarB/AQFT/LocalNet.lean	CNNA/AQFT/LocalNet.lean	direkt portieren und später sektorisieren
REALOQS/PillarB/AQFT/GNS.lean	CNNA/AQFT/GNS.lean	direkt portieren; ComplementGNS darauf aufbauen
REALOQS/PillarB/AQFT/KMS.lean	CNNA/AQFT/KMS.lean	direkt portieren; ComplementKMS darauf aufbauen
REALOQS/PillarB/AQFT/RelativeEntropy.lean	CNNA/AQFT/RelativeEntropy.lean	direkt portieren
REALOQS/PillarC/OQS/Channel.lean	CNNA/OQS/SectorChannels.lean	semantisch verbreitern von System-Umgebung auf Sektor-Kanäle
REALOQS/PillarC/OQS/Stinespring.lean	CNNA/OQS/Stinespring-kompatibleUnterlage	direkt als mathematische Grundschrift behalten

### 7.2 Module, die semantisch neu gelesen werden müssen

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

Diese Gruppe ist der eigentliche Migrationsmotor. Hier reicht kein Umbenennen; die begriffliche Achse muss vom binären boundary-first Split auf die Mehrsektor- und Interface-Sprache gedreht werden.

Tabelle 19: Module, die semantisch neu gelesen werden müssen – Teil 1

Altmodul	Ziel in CNNA	Disposition
REALOQS/PillarA/OQS/SysEnv.lean	CNNA/OQS/SectorSysEnv.lean	binären Split durch Mehrsektor-Split ersetzen
REALOQS/PillarA/Update/TailEliminationDtN.lean	CNNA/OQS/MultiSchur.lean+CNNA/Integration/TailElimAsChannel.lean	Tail-Elimination nicht nur eliminatorisch, sondern kanalisch lesen
REALOQS/PillarA/Core/TailEliminationCoherence.lean	CNNA/Core/ParameterClosure.lean+CNNA/OQS/RegularizationClosure.lean	Koherenz in Closure-/Backreaction-Sprache überführen
REALOQS/PillarC/Integration/DerivedSpacetime.lean	CNNA/PillarD/DerivedSpacetime.lean	von C-Endpunkt zu D-Vorbereitungsobjekt umdeuten
REALOQS/PillarC/Integration/Locality.lean	CNNA/PillarD/InterfaceCausality.lean+CNNA/PillarD/LocalCovariance.lean	lokale Einflussrelation in lokale Kovarianz überführen

### 7.3 Was nur noch als Bright-Spezialfall dienen darf

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

Gerade die stärksten alten Derived-Dateien sind für Regression und Bright-Recovery wichtig. Sie dürfen aber die neue Architektur nicht semantisch dominieren.

Tabelle 21: Was nur noch als Bright-Spezialfall dienen darf – Teil 1

Altmodul	Status	Folge
REALOQS/PillarA/Ideal/Adapter/TreeOfCliquesAQFTHandoff.lean	Bright-Spezialfall / Regressionsorakel	nicht Architekturzentrum, sondern Vergleichspfad
REALOQS/PillarB/AQFT/Derived/TreeOfCliquesFromPillarA.lean	Bright-Spezialfall / Regressionsorakel	nur replays, nicht zentrieren
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixLocalNet.lean	Bright-Spezialfall unter CNNA/AQFT	alte Randnetz-Semantik als abgeleitete Restriktion rekonstruieren
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixStateNet.lean	Bright-Spezialfall unter CNNA/AQFT	nur als Restriktion auf Bright
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixGNS.lean	Bright-Spezialfall unter CNNA/AQFT	nur als Restriktion auf Bright
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixFullKMSFromL.lean	Bright-Spezialfall unter CNNA/AQFT	nur als Restriktion auf Bright
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixQuasiLocalClosure.lean	Bright-Spezialfall / Regression	keine architektonische Zentralstellung
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixModular.lean	Bright-Spezialfall / Regression	modulare Daten später aus CNNA zuruckgewinnen

### 7.4 Späte Seeds und diagnostische Inputs für D und E

Dokumentenebene: Projekt-Ebene und Pfad-Ebene

Diese Module sind wertvoll, aber nur als Input, Hook oder späte diagnostische Spur. Sie dürfen vor der jeweiligen Closure-Schicht nicht als schon geschlossene Ontologie der neuen Pfeiler missverstanden werden.

Tabelle 23: Späte Seeds und diagnostische Inputs für D und E – Teil 1

Input aus v0.0605	Ziel in D/AS/E	Regel
REALOQS/PillarA/Kinematics/DiscreteSpacetime.lean	CNNA/PillarD/DerivedSpacetime.lean	als Seed-Herkunft, nicht als fertige Raumzeit
REALOQS/PillarA/Kinematics/Causality.lean	CNNA/PillarD/InterfaceCausality.lean CNNA/PillarD/SpectralDimensionFlow	nur nach expliziter Interface-Herkunft de-seeden
REALOQS/PillarA/Geometry/DimensionHooks.lean	Seed.lean CNNA/PillarD/HorizontalLevelSeed.lean	zunächst Hook/Seed
REALOQS/PillarA/RG/ScaleWindow.lean	und AS-Strang	Skalenfenster nicht ontologisch, sondern als derived Diagnose lesen
REALOQS/PillarA/Physics/ScaleBreaking*.lean	CNNA/PillarD/Backreaction-undClosure-Diagnostik	nur späte Diagnostik, kein Kernaxiom

**Harte Migrationsregel.** Kein boundary-first Dateiname, kein freies  $L_{\max}$ , kein freier State-, Dictionary- oder Observer-Begriff und kein unmarkiertes Spezialfenster darf auf dem kritischen CNNA-Pfad als echter Kernbegriff erscheinen. Diese Regel folgt direkt aus der internen Spezifikation.[2, pp. 2–9]

## 8 Pillar A – ontische Stammtheorie und Komplementarchitektur

**Dokumentebene:** Pfad-Ebene

### 8.1 Zielkern von A

**Dokumentebene:** Pfad-Ebene

Pillar A muss in CNNA den ontischen Stamm liefern: ToC, Anker, sichtbarer Patch, Komplementfamilie, Mehrsektor-Split, verallgemeinerte DtN-/Schur-Mechanik und erste Parameter-Closure. A ist in der neuen Architektur nicht bloss Lieferant eines Randoperators, sondern Generator der Sektorgrammatik selbst. Das Snowmass-Argument gegen eine einzig privilegierte Axiomatik stärkt genau diesen Schritt: CNNA darf nicht auf der ersten mathematisch bequemen Oberfläche stehenbleiben.[4, pp. 1, 10–12]

### 8.2 Konkrete Vorgehensweise in A

**Dokumentebene:** Pfad-Ebene

**1. Tragerkern einfrieren und bereinigen.** Portiere `Approximant`, `RegionCore`, `RegionNet`, `InfiniteCarrier`, `Selection`, `Interfaces`, `Determinism`, `BoundaryPorts`, `DirichletLaplacian` und `DirichletLaplacianBridge` in einen neuen CNNA/Core- und CNNA/ToC-Baum. In diesem ersten Schnitt werden nur diejenigen Deklarationen übernommen, deren mathematische Bedeutung ohne boundary-first Zusatzsprache stabil bleibt. Alles, was implizit bereits den alten Bright-Randkanal privilegiert, wird aus dem aktiven Pfad entfernt oder markiert.

**2. BranchPatch und Komplementfamilie als erste Klasse einführen.** Schreibe `CNNA/Core/BranchPatch.lean` und `CNNA/Core/ComplementSectorFamily.lean` nicht als freie Container, sondern als direkte Verankerung an ToC-Anker, Pfadstruktur, Zellen und Koarsenierungsdaten. Der alte `Approximant` ist nur der helle Ausschnitt; der dunkle Sektor entsteht aus einer *organisierten* Komplementfamilie, nicht aus  $S \setminus R$  als namenlosem Rest. Damit verschiebt sich der Grundbegriff des Projekts von `Boundary` zu `Sector`.

**2a. ToC-Varianten nur über einen gemeinsamen CNNA-Vertrag zulassen.** Die Roadmap sollte hier nicht nur verbal, sondern mathematisch explizit bleiben: der ToC ist im CNNA-Kontext kein beliebiger Graphname, sondern ein Trager mit Ankern, Pfad-/Trunk-/Side-Branch-Information, lokaler Inzidenz, Koarsenierung und Source-Maps in Richtung Split- und Kopplungsschicht. Genau diese Daten müssen später `BranchPatch`, `ComplementSectorFamily`, `SectorSplit` und `GeneralizedDtN` speisen. Diskrete, kontinuierliche oder gradierte Varianten des ToC sind daher nur als *Varianten eines gemeinsamen Vertrags* zulässig: sie dürfen Topologie, Regularität oder Zusatzsymmetrien ändern, müssen aber dieselben sektorbezogenen Witness- und Restriktionspflichten erfüllen. Die Frage ist also nicht “welcher ToC-Stil gefällt”, sondern ob eine Variante die Split- und Komplementlogik funktoriell trägt.

**2b. Prototypische B-/C-Schnittstellen früh als Seeds anlegen.** Um Typ-Sackgassen zu vermeiden, darf A dem späteren B-/C-Pfad kontrolliert vorauslaufen. Konkret heißt das: neben den eigentlichen Kernobjekten dürfen frühe `Sketch`-Dateien oder `Seed`-Schnittstellen für spätere Verbraucher mitlaufen, etwa für `ComplementLocalNet`-Signaturen, sektorweise Restriktionen, `Glueing`-Daten, komplementbasierte Zustandsprojektionen oder einfache Platzhalter für modulare Aktionen. Solche Deklarationen sind gerade *nicht* die vollständige Mathematik von B oder C; sie müssen als `PrototypeInterface`, `Seed` oder `Scaffold` markiert bleiben und werden nur daran gemessen, ob sie die A-seitigen Typen, Exporte und Indizes später tragfähig machen.

Was auf diesem Weg nur mit ad-hoc Koerzionsketten, unsauberen Universumsabkürzungen oder boundary-first Sonderannahmen compilieren würde, ist ein Frühwarnsignal gegen den aktuellen A-Vertrag.

**3. Binären Split durch Mehrsektor-Split ersetzen.** Aus `SysEnv`- und `SplitInvariants`-artigem Material wird ein neuer `SectorSplit` aufgebaut. `Bright`, `Dark` und `Interface` müssen gemeinsam auftreten; insbesondere muss das `Interface` als eigener Träger modelliert werden, nicht als implizite Grenzspur zwischen zwei anderen Datenblöcken. Genau an dieser Stelle wird später die Glueing-, Entropie- und Defektlogik aufgesetzt.

**4. GeneralizedDtN und MultiSchur als mathematischen Kopplungskern bauen.** Der alte `DtN`-Pfad wird nicht entsorgt, sondern als binärer Grenzfall verstanden. Die mathematische Strategie auf dem aktiven Pfad ist jedoch strenger: zürst eine wohldefinierte Mehrsektor-Blocksicht auf `Bright`, `Dark` und `Interface` herstellen, dann globalen und iterierten `Schur`-/`DtN`-Abbau gegeneinander beweisen und erst danach spezielle Stabilisierung, Positivitäts- oder Regularisierungsfragen aufsetzen. Genau dieser Vergleich von globaler und iterierter Elimination — architektonisch repräsentiert durch `MultiSchur` und mathematisch zugespitzt im Zieltyp von `iteratedSchur_eq_globalSchur` — ist das nichttriviale Kerntheorem des `GeneralizedDtN`- Strangs; er darf nicht hinter spätere Spezialfälle oder numerische Hilfskonstruktionen ausweichen.

Im CNNA-Kern ist `GeneralizedDtN` algebraisch stärker als eine einzelne Randmatrix: gebraucht wird mindestens eine sektorindexierte Block- bzw. Operatorkernstruktur, die `Bright`-, `Dark`- und `Interface`-Komponenten zugleich koppelt, `Schur`-Reduktion und Restriktion entlang von Teilsektoren zulässt und mit Symmetrie-/Positivitätszeugen auftritt, statt nur eine ad-hoc Matrix zu liefern. Genau dieser Kern muss die spätere `ComplementLocalNet`-Schicht speisen: lokale Algebren dürfen nicht losgelöst vom Kopplungskern entstehen, sondern müssen aus sektorweisen Ports, Restriktionen und `MultiSchur`-Reduktionen hervorgehen. Lean-seitig spricht das klar für eine `structure`-basierte Darstellung mit expliziten Feldern für Sektorindex, Kern, Restriktion, Glueing und Kompatibilitätszeugen, während Typeclasses und kategoriale Hülle erst nachgeordnet für Instanzen, Morphismen oder Funktorialität sinnvoll sind. Dazu gehört auch, `MultiSchur.lean` als eigenes Zielmodul sichtbar zu halten, statt den global-vs.-iteriert-Vergleich implizit in Hilfslemmata zu verstecken.

**5. Parameter-Closure bereits in A vorbereiten.** Freie Strukturparameter aus dem Altpfad — vor allem  $b$ ,  $1 < b$ , Horizon-/IR-Größen,  $L_{\max}$  und sekundäre numerische Stabilisierungsterme — dürfen nicht unreflektiert in CNNA übernommen werden. In A ist deshalb früh eine `BranchingWitness`- und `ParameterClosure`-Schicht zu schreiben, die drei Ebenen sauber trennt: ontische Mikrostrukturparameter wie  $b$ , dynamische Emergenzgrößen wie  $L_{\max}$  bzw. später `HorizonLevel`, sowie bloss numerische Regularisierer. Für  $b$  bedeutet das: nicht nur  $1 < b$  als Nichttrivialitätsmarker festhalten, sondern über `UVSpectralSelector`, `BranchingSelector` und `BackreactionSelector` einen späteren kombinierten Selektorpfad bis zu einer abgeleiteten `selectedBranching`-Lesart vorbereiten. Für  $L_{\max}$  bedeutet es, dass der Name auf dem aktiven Pfad höchstens Scaffold-Status hat und später durch `HorizonLevel` bzw. einen `BackreactionFixedPoint` ersetzt wird. Für Regularisierer bedeutet es schliesslich, dass sie entweder ganz verschwinden oder als abgeleitete Spektral-/Nullmodenreste in eine `RegularizationClosure` degradiert werden. [2, pp. 2, 5–9, 29–31, 44–47]

**6. Exporte neu schreiben, nicht den alten Handoff verabsolutieren.** Der alte `AQFTHandoff`-Gedanke bleibt wichtig, aber sein Output darf nicht wieder nur  $(\Omega, L, \beta)$  für einen `Bright`-Rand sein. Der neue Export aus A muss die Sektortriple (`Bright`, `Dark`, `Interface`) samt `Split`-, `GeneralizedDtN`- und `Source-Map`-Daten bereitstellen. Erst darauf dürfen B und C aufsetzen.

### 8.3 Abnahmekriterien für A

**Dokumentebene:** Pfad-Ebene

A ist erst dann erfolgreich migriert, wenn:

- `BranchPatch`, `ComplementSectorFamily`, `SectorSplit` und `GeneralizedDtN` als echte Kernobjekte vorliegen;
- der alte binäre DtN-Fall als Sonderfall erklärbar ist;
- $L_{\max}$  auf dem aktiven Pfad nicht mehr ontisch gebraucht wird;
- der Export aus A sektorisch und nicht boundary-first formuliert ist;
- prototypische B-/C-Schnittstellen auf A-Basis ohne ad-hoc Typreparaturen, boundary-first Sonderannahmen oder versteckte freie Beobachter kompiliert werden können;
- ein detaillierter B-Phasen- bzw. Unterplan erst dann als belastbar gilt, wenn das A-Gate nicht nur mathematisch, sondern auch interface-stabil gegen diese Vorgriffe ist.

Als operative *Definition of Done* für den Übergang zu detaillierter B-Planung gilt damit: `BranchPatch`, `ComplementSectorFamily`, `SectorSplit`, `GeneralizedDtN` und sektorischer Export stehen; zugleich kompilieren die frühen `PrototypeInterface`-Skizzen für Netz-, Zustands- und Glueing-Verbraucher ohne semantische Sonderhaken. Erst auf dieser Lage wird aus B-Planung mehr als eine reversible Absichtserklärung.

### 8.4 Was in A strikt zu vermeiden ist

**Dokumentebene:** Pfad-Ebene

- direkte Portierung der alten boundary-first Dateinamen als Zentrum des neuen Modulbaums;
- Einschmuggeln eines freien Beobachters;
- frühe numerische Stabilisierung als Ersatz für hergeleitete Struktur;
- Bright-Recovery schon in A selbst als globales Architekturziel.

## 9 Pillar B – AQFT aus dem Komplementnetz statt aus dem Bright-Rand

**Dokumentebene:** Pfad-Ebene

### 9.1 Zielkern von B

**Dokumentebene:** Pfad-Ebene

Pillar B muss aus der in A bereitgestellten Sektorstruktur ein wirkliches Komplementnetz bauen: `ComplementLocalNet`, `InterfaceLocalNet`, `ComplementStateNet`, `ComplementGNS`, `ComplementKMS`, dazu die quasilokale und Haag-Kastler-nahe Unterlage. Snowmass liest AQFT, lokal kovariante QFT, homotopy AQFT, Functorial Field Theory und CFT gerade als unterschiedliche QFT-Faces; CNNA muss deshalb in B die AQFT-Schicht so bauen, dass sie später für mehrere Faces offen bleibt.[4, pp. 4–12]

### 9.2 Konkrete Vorgehensweise in B

**Dokumentebene:** Pfad-Ebene

**1. Die generische AQFT-Basis unverzerrt portieren.** `StarAlgebra`, `State`, `StateNet`, `LocalNet`, `GNS`, `KMS`, `RelativeEntropy`, `QuasiLocal/*` und die relevanten Haag-Kastler-Bausteine werden zürst mathematisch sauber nach CNNA/AQFT portiert. Wichtig ist, dass diese Grundschrift nicht stillschweigend Bright-spezifische Matrix-Semantik erbt.

**1a. Die bestehende Gate-Schicht aus B explizit mitportieren.** Die 20 Dateien unter `REALOQS/PillarB/Gates` sind keine dekorative Audit-Prosa, sondern die formale Validierungsschicht für `StarAlgebra`-, `Norm`-, `StateRestriction`-, `QuasiLocal`- und Haag-Kastler-Aussagen. In CNNA werden sie daher als eigene Gate-Ebene neu aufgesetzt: `CNNA/Gates/AQFT/*`, `CNNA/Gates/HaagKastler/*`, `CNNA/Gates/ComplementAdditivity.lean`, `StateRestriction.lean`, `SplitProperty.lean` und funktoriale Netzgates. Besonders die Teilgates für Isotonie, Lokalität, Additivität, starke Additivität, Kovarianz, Spektrum, Time-Slice und Vakuum dürfen nicht in allgemeiner Prosa verschwinden, weil sie später die Abnahme des Komplementnetzes gegen den Bright-Recovery-Pfad definieren.[5][30]

**1b. Die algebraische Grundschrift von Anfang an Z2-offen anlegen.** Die direkte Portierung von `StarAlgebra.lean` bleibt richtig, reicht für die spätere Spin-Statistik-Schicht aber nicht aus. Sobald CNNA bosonische und fermionische Sektoren sauber unterscheiden will, braucht die B-Grundschrift mindestens eine optionale Z2-Graduierung mit Superkommutator, Paritätsmarkierung und kontrollierter Rückbindung an die ungraduierte Sternalgebra. Deshalb wird früh zusätzlich `CNNA/AQFT/GradedStarAlgebra.lean` geplant. Das ist kein später Luxus, sondern die Minimalvorkehrung dafür, dass `FermionicStatisticsSeed`, `StatisticsGate`, `SpinRepresentationSeed` und später `KramersGate` nicht als freie Zusatzetiketten über einer rein bosonisch gelesenen Algebra schweben. Die mathematische Richtung ist durch die klassische Spin-Statistik-Literatur und ihre algebraische AQFT-Fortsetzung klar vorgegeben. `GradedStarAlgebra` und `SeparatingProperty` sind dabei bewusst *logisch unabhängig*: die erste Datei betrifft die algebraische Form der Observablenbasis, die zweite die spätere Darstellungslage des Komplementnetzes. Weder darf daher eine künstliche Importkette `GradedStarAlgebra->SeparatingProperty` behauptet werden noch umgekehrt. [41][42]

**2. ComplementLocalNet vor jeder Recovery schreiben.** Der erste echte Netzbau in B muss das dunkle Netz sein, nicht die Wiederholung des alten `BoundaryMatrixNet`. Konzeptuell heißt das: Regionen aus `BranchPatch/ComplementSectorFamily/SectorSplit` ableiten,

Algebren aus blockweiser Operator- und GeneralizedDtN-Struktur zuweisen, danach Isotonie und Lokalität beweisen. Erst wenn dieser Schritt trägt, darf ein Bright-Spezialfall überhaupt als Regression dienen.

**3. InterfaceLocalNet als eigenen mathematischen Gegenstand behandeln.** Die größte architektonische Versuchung wäre, das Interface nur als technische Zwischenzone zu benutzen. Das wäre strategisch falsch. Das Interface-Netz ist die Stelle, an der später Glueing-Funktionalität, Defekte, Entropie, thermische Balance und Dualitätsdaten gesammelt werden. Deshalb sollte `CNNA/AQFT/InterfaceLocalNet.lean` parallel zu `ComplementLocalNet.lean` geschrieben werden, nicht als späte Nachbesserung.

**4. Zustandsanschluss strikt aus Sektorrelationen gewinnen.** `ComplementStateNet` darf kein freier `State`-Träger sein. Zustände müssen als Restriktionen, Pushforwards oder Balance-Zustände auf Bright, Dark und Interface erscheinen. Erst wenn diese Restriktionen funktionieren, lohnt sich der Schritt zu `ComplementGNS` und `ComplementKMS`. Thermik und Modularität gehören also in B an das Ende des sektorischen Zustandsaufbaus, nicht an dessen Anfang.

**4a. Zwischen ComplementGNS und ComplementKMS eine echte modulare Kernschicht einfügen.** Die im Archiv vorhandene Bright-Recovery-Datei `BoundaryMatrixModular.lean` ist als Regression nützlich, aber sie extrahiert weder die Tomita-Operation noch modulare Konjugation  $J$  und modularen Operator  $\Delta$  als eigenständige CNNA-Objekte. Für die neue Architektur reicht das nicht. Nach `ComplementGNS` muss daher eine Reeh-Schlieder-artige Separationsschicht folgen, die den GNS-Vektor nicht nur als zyklisch, sondern für die relevanten Komplementalgebren auch als separierend kontrolliert. Erst darauf dürfen `CNNA/AQFT/SeparatingProperty.lean`, `CNNA/AQFT/TomitaTakesakiData.lean` und `CNNA/AQFT/ModularConjugation.lean` aufsetzen. Der Zielpunkt ist nicht blosse Terminologie, sondern ein Theoremstil der Form `modular_conjugation_maps_bright_to_complement`, also  $JMJ = M'$  in der für CNNA passenden Bright/Dark/Interface-Lesart. Genau diese Stufe gehört in den B-Kern, nicht in eine späte Matter-Vorzone.[37][38][42]

**4b. Die modulare B->D-Brücke explizit als Bisognano-Wichmann-Seed markieren.** Sobald Tomita-Takesaki-Daten vorhanden sind, darf CNNA den Übergang nach D nicht nur heuristisch über “Entropie” oder “Kausalität” erzählen. Vor dem eigentlichen Bisognano-Wichmann-Vergleich muss dabei zusätzlich eine explizite Spektralkontrolle des *Komplementnetzes* vorliegen: `CNNA/AQFT/ComplementSpectrumCondition.lean` sammelt Vorwärtskegel-, Positivitäts- und Translationsdaten auf dem aktiven `derived-only` Pfad, und das Gate `gate_spectrum_condition_on_complement_net` stoppt jede spätere Guido-Longo-, Bisognano-Wichmann- oder Haag-Ruelle-Lesart, solange diese Kontrolle nur aus Bright-Recovery importiert wäre. Erst darauf darf mit `CNNA/AQFT/BisognanoWichmannSeed.lean` eine eigene Brückendatei als *spätes B-Modul* folgen, die modulare Automorphismen mit abgeleiteten Boost- bzw. Wedge-Strukturen vergleicht. Der Zielstil ist `modular_automorphism_group_equals_derived_boost_on_wedge_sector`. Die Stop-Regel des Seeds selbst lautet dabei: *nach* `ComplementKMS` und `ComplementSpectrumCondition`, *vor* den Pfeilerübergreifenden PT/CPT-Brückenobjekten; der spätere Abgleich mit `InterfaceCausality` wird *nicht* als Importvoraussetzung des Seeds selbst kodiert, sondern erst über das Gate `gate_modular_boost_consistent_with_interface_causality` vollzogen. So wird die Relation zwischen B-Modularität und D-Geometrie physikalisch erzwungen statt blos architektonisch behauptet, ohne einen  $B \leftrightarrow D$ -Zirkel in die Dateitopologie einzubauen.[39][40][42][29]

### 9.3 B-Kernformalismus der modularen Darstellungsarchitektur

**Dokumentebene:** Pfad-Ebene

Der relevante B-Kern endet deshalb nicht bei einem losen Paar `ComplementGNS/ComplementKMS`. Die operative Kette ist strikt:

**Tabelle 25: B-Kernformalismus der modularen Darstellungsarchitektur – Teil 1**  
`ComplementStateNet` → `ComplementGNS` → `SeparatingProperty`  
 → `TomitaTakesakiData` → `ModularConjugation`  
 → `ComplementKMS` → `ComplementSpectrumCondition`  
 → `BisognanoWichmannSeed`.

Die ersten fünf Pfeile gehören in den B-Kern selbst. `ComplementGNS` allein reicht nicht, weil ohne Separierung weder Tomita–Takesaki noch eine kontrollierte modulare Konjugation verfügbar sind; `ComplementKMS` ist dann kein Ersatz für diese Schicht, sondern ihr thermischer Verbraucher. Erst der anschliessende `BisognanoWichmannSeed` schliesst den späten modularen B-Unterpfad und macht die modulare Struktur für D als Geometrie-/Kausalitätsbrücke operational; die Konsistenz mit `InterfaceCausality/LocalCovariance` wird anschliessend über das eigene Gate `gate_modular_boost_consistent_with_interface_causality` geprüft. Die operative Feingliederung dieser Kette als Schritte 6a–6i wird in Abschnitt 13.2 zusammengestellt. Die Nuclearity-Linie beginnt dabei *nicht* schon vor `ComplementKMS`, sondern streng danach: sie ist B-intern, aber eine Nach-KMS-Schicht auf bereits geschlossener modularer Darstellungsarchitektur.

## 9.4 Warum das Komplementnetzwerk für full-derived lokale QFT physikalisch zwingend wird

**Dokumentebene:** Pfad-Ebene

Die Roadmap muss hier eine asymmetrische, aber wichtige Präzisierung festhalten. Ein einzelnes Kramers-Theorem in endlicher Quantenmechanik oder eine isolierte Antiunitaritätsaussage lässt sich auch ohne voll ausgearbeitete Komplementarchitektur formulieren. CNNA zielt jedoch nicht auf lose Einzelsätze, sondern auf eine gemeinsame Rekonstruktion von lokaler Algebra, Zustand, Darstellung, Modularität, Kausal-/Geometriestruktur und daraus erst CPT-, Spin-Statistik- und Kramers-Lesarten. In genau diesem physikalischen Sinn wird das Komplementnetzwerk weitgehend zwingend.

Erstens ist das bereits durch den derived-only Kern der Spezifikation vorgegeben: Regionen, Komplementfamilien, Interface-Kanäle, Zustandsrestriktionen und Backreaction sollen nicht als sekundäre Dekoration auftreten, sondern als Herkunftsträger der späteren Physik. `ComplementSectorFamily`, `SectorSplit`, `ComplementLocalNet`, `ComplementStateNet`, `ComplementGNS` und `ComplementKMS` bilden daher nicht nur einen alternativen Pfad, sondern die Stelle, an der aus Bright/Dark/Interface überhaupt eine physikalisch belastbare lokale Theorie emergiert.[2, pp. 1–10, 30–36]

Zweitens hängt gerade die angestrebte vN-/Typ-III-Lesart nicht an einer isolierten Matrixalgebra, sondern an der Verschänkung von Lokalität, Darstellung und modularer Komplementarität. Ohne separierende Vektoren, Tomita–Takesaki-Daten und die Bright/Dark/Interface-Lesart von  $JMJ = M'$  bliebe höchstens ein Typ-I-naher Testträger übrig; die für lokale AQFT relevante Komplementstruktur wäre dann weder algebraisch noch physikalisch geschlossen.[38][42]

Drittens sitzen CPT, Spin-Statistik und Kramers in der hier angestrebten Lesart gerade nicht auf einem einzelnen Hamiltonoperator, sondern auf dem Zusammenspiel von lokaler Algebra, Darstellungsraum, Spektrum, Antiunitarität, modularer Struktur und später Kausal-/Geometrie-Information. Ohne komplementbasierte Darstellungs- und Modularschicht könnten diese Sätze zwar formal irgendwo nachgebildet werden; sie wären im CNNA-Sinn aber nicht mehr *aus derselben Stammtheorie* rekonstruiert.[39][40][42]

Die operative Designlesart lautet deshalb scharf: Ohne Komplementnetz keine echte full-derived lokale QFT-Physik; und ohne komplementbasierte Darstellungs- und Modularstruktur bleiben CPT, Spin-Statistik und Kramers in CNNA entweder unvollständig oder bloss aufgesetzt. Genau deshalb gehört die Komplementseite nicht in einen späten Appendix, sondern in den tragenden Kernpfad von B.

**5. Vakuumfrage, Grundzustände und derived-only Ersatzbegriffe.** Auch die Rolle des Vakuums muss im Roadmap-Text explizit geklärt werden. Im alten Bright-Recovery- Pfad existiert ein konkreter Grund- bzw. Vakuumkandidat als Grenz- oder Niedrigtemperaturlesart des bekannten Zustandsaufbaus. Im CNNA-Kern gibt es hingegen *kein* freies Vakuumpostulat: die aktive Architektur verlangt, dass relevante Zustandslagen aus Komplement-/Interface-Daten, Restriktionen und Balancebedingungen hervorgehen. Die `SeparatingProperty` des GNS-Vektors übernimmt hier die Rolle eines abgeleiteten Vakuumsatzes für die modulare AQFT-Lesart; auf späteren emergenten Hintergründen in D wird diese Sonderrolle nicht durch ein global ausgezeichnetes Vakuum, sondern durch die Hadamard-/mikrolokale Zulässigkeitsklasse ersetzt. Gerade diese Verschiebung von postuliertem Vakuum zu abgeleiteter Zustands- und Separationslage ist für eine strictly derived-only Lesart entscheidend.[30][37][6]

**5a. Kontinuums- und thermodynamischen Limes explizit markieren.** Der Bright-Approximant bleibt endlich; die spätere AQFT-Lesart von quasilokaler Algebra, Tomita–Takesaki, Typ-III-Struktur, Bisognano–Wichmann und Haag–Ruelle ist es gerade nicht. Darum muss auf dem aktiven Pfad zwischen endlichem Patch und unendlicher/quasilokaler Darstellungslage ein expliziter Übergang via `InfiniteCarrier`, quasilokaler Vervollständigung oder thermodynamischer Limesmarkierung auftauchen. Dieser Limes ist nicht bloss technische Nachpolitik, sondern stoppt den Pfad, solange unendliche Kommutanten- und Modularstruktur nur aus endlichen Bright-Matrizen gelesen würde. Offen bleiben darf, *welcher* operative Schliessungsweg am Ende gewinnt; offen bleiben darf aber nicht, *dass* zwischen endlichem Approximanten und AQFT-Lage ein eigener Schritt liegt. Die Roadmap sollte deshalb drei zulässige Alternativpfade benennen: (i) *InfiniteCarrier by construction*, falls das aktive Komplementnetz bereits als unendliche/quasilokale Struktur aufgebaut wird und der endliche Bright-Patch nur Restriktionsfenster bleibt; (ii) *induktive/quasilokale Vervollständigung* aus einer gerichteten Familie wachsender `BranchPatch`- bzw. Sektor-Ausschnitte; (iii) *thermodynamischer/KMS-Limes* mit expliziten uniformen Spektral-, Positivitäts- und Modularitätskontrollen. Nicht feststehen muss heute, welche Route gewinnt; feststehen muss aber, dass mindestens eine dieser Routen samt Gate `gate_infinite_carrier_or_quasilocal_limit_defined` wirklich geschlossen wird, bevor Typ-III-, BW- oder Scattering-Lesarten als derived beansprucht werden.

**5b.  $\beta$ , Unitarität und Positivität nicht als freie Restparameter stehenlassen.** Im alten Bright-Pfad ist  $\beta$  ein gegebener Input des Gibbs-/KMS-Aufbaus. Auf dem aktiven CNNA-Pfad darf das nicht das Endbild bleiben. Entweder wird  $\beta$  später aus modularer Automorphismengruppe und ihrer physikalischen Zeitnormalisierung abgeleitet — die stärkste Lesart entsteht genau dann, wenn Bisognano–Wichmann den modularen Fluss an emergente Geometrie bindet — oder  $\beta$  bleibt bis zur `BackreactionFixedPoint`-Schicht ein explizit markierter Scaffold- bzw. Fixpunktparameter. Analog dürfen Unitarität der effektiven Entwicklung und positive Energie-/Vorwärtskegelkontrolle nicht nur stillschweigend vorausgesetzt werden: `ComplementSpectrumCondition` trägt die Positivitätsseite für das Komplementnetz, während C und D eine explizite Kontrolle benötigen, dass ihre physikalisch interpretierten Flüsse und Generatoren nicht in unkontrollierte Nichtunitarität oder negative-Energie-Heuristiken kippen.[30][39][40]

## 9.5 Zwingende AQFT-Modulergänzungen im Modulbaum

**Dokumentebene:** Pfad-Ebene

Die modulare Zwischenschicht muss im B-Modulbaum ausdrücklich und nicht nur implizit auftauchen. Für den kritischen AQFT-Unterbaum sind daher mindestens die folgenden Dateien zwingend:

**Tabelle 26: Zwingende AQFT-Modulergänzungen im Modulbaum – Teil 1**

Zieldatei	Rolle im B-Kern
<code>CNNA/AQFT/GradedStarAlgebra.lean</code>	optionale, aber früh verankerte Z2-Graduierung der algebraischen Grundschicht; keine fermionische Semantik vor dieser Datei
<code>CNNA/AQFT/SeparatingProperty.lean</code>	Reeh-Schlieder-artige Separationsstufe zwischen <code>ComplementGNS</code> und modularer Maschine
<code>CNNA/AQFT/TomitaTakesakiData.lean</code> <code>CNNA/AQFT/ModularConjugation.lean</code>	extrahiert $S$ , $J$ und $\Delta$ aus der komplementnetz-basierten Darstellung kontrolliert die Bright/Dark/Interface-Lesart von $JMJ = M'$ als eigentliche modulare Darstellungsarchitektur
<code>CNNA/AQFT/ComplementSpectrumCondition.lean</code>	explizite Spektral- und Vorwärtskegelkontrolle des Komplementnetzes vor jeder BW-, Guido-Longo- oder Haag-Ruelle-Lesart
<code>CNNA/AQFT/BisognanoWichmannSeed.lean</code>	modulare $B \rightarrow D$ -Brücke; verknüpft Wedge-/Boost-Strukturen mit modularen Automorphismen

Diese Dateigruppe ist *kein* später E-Zusatz. Sie gehört in den AQFT-Unterbaum selbst und schliesst die Lücke zwischen Zustand, Darstellung, Modularität und Geometrie. Die späteren Symmetrie-/Statistik-Dateien wie `TimeReversalOnGNS`, `AntiunitarySeed`, `SpinRepresentationSeed`, `FermionicStatisticsSeed`, `CPTGate` und `KramersGate` konsumieren diese B-Kernschicht, ersetzen sie aber nicht.

**6. Die alten BoundaryMatrix-Dateien nur als Extraktionsquelle verwenden.** Aus den alten Derived-Dateien werden genau diejenigen Teile herausgelöst, die wirklich allgemein sind: lineare Algebra, allgemeine Netztechnik, Restriktions- oder Closure-Lemmata. Alles ToC-gebundene am hellen Rand wird explizit in einen Bright-Recovery-Strang verschoben. Damit bleibt die mathematische Arbeit der alten Dateien nutzbar, ohne dass deren Semantik das neue Projekt dominiert.

**7. Quasilokale und HK-nahe Zusatzstrukturen erst nach Netzstabilität anschliessen.** Sobald dunkles Netz, Interface-Netz und Zustandsrestriktionen tragen, wird die quasilokale Ergänzung sowie die HK-nahe Support-/Split-/StateRestriction-Schicht angeschlossen. Für diesen Schritt sind die Snowmass-Hinweise auf AQFT, LCQFT, dynamical locality und spätere homotopische Verallgemeinerungen wichtig: CNNA sollte hier strukturell offen bleiben.[4, pp. 4–12][5][6][7][8]

**8. Superselection-, Charge- und Feldrekonstruktionsschicht explizit vorsehen.** Spätestens auf stabiler HK-/Quasilokal-Unterlage braucht CNNA eine observablenseitige Superselection-  
`CNNA/AQFT/FieldAlgebraReconstruction`  
Schicht. Deshalb werden `CNNA/AQFT/DHRsectorsSeed.lean` und `Seed.lean` ausdrücklich eingeplant: erst lokalisierte DHR-Sektoren, dann eine kontrollierte DR-artige Rekonstruktion von Feldalgebra und kompakter Eichgruppe aus der Sektorstruktur. Dadurch bleiben `ChargeConjugationSeed`, `GaugeSectorSeed` und spätere Statistik-/Teilchenrede auf die Observablen- und Netzschicht rückgebunden, statt als freie Seeds zu schweben.[24][25]

**9. Split Property nicht als Endpunkt missverstehen: Nuclearity als eigene Schicht führen.** Die bereits geplante Split-Property bleibt wichtig, reicht aber als Fernziel nicht aus. Für kontrollierte Phase-Space-Dichte, thermische Belastbarkeit und spätere Typ-III-Lesarten wird zusätzlich eine Nuclearity-Schicht benötigt. Deshalb werden `CNNA/AQFT/NuclearitySeed.lean` und `CNNA/Gates/Nuclearity.lean` explizit neben Split/Support geführt und nicht als später optionaler Kommentar behandelt.[26]

## 9.6 Abnahmekriterien für B

Dokumentenebene: Pfad-Ebene

B ist erst dann tragfähig, wenn:

- `ComplementLocalNet` und `InterfaceLocalNet` die ersten nichttrivialen Netzobjekte sind;
- Isotonie, Lokalität und Zustandsrestriktion für diese Netze explizit formuliert sind;

- die Grundschrift entweder bereits `GradedStarAlgebra` trägt oder mindestens explizit `Z2`-offen für spätere fermionische Sektoren angelegt ist;
- `ComplementGNS` nicht isoliert bleibt, sondern durch `SeparatingProperty`, `TomitaTakesakiData` und `ModularConjugation` bis zur modularen Kernschicht ergänzt ist;
- `ComplementKMS` auf dieser Zustands- und Modularity-Schicht aufsetzt statt sie zu ersetzen;
- der Übergang vom endlichen Bright-Approximanten zu `InfiniteCarrier`, quasilokaler Vervollständigung oder thermodynamischer Limeslage explizit markiert ist, bevor unendliche AQFT-Lesarten beansprucht werden;
- eine explizite Spektralkontrolle des aktiven Komplementnetzes via `ComplementSpectrumCondition` bzw. `gate_spectrum_condition_on_complement_net` vorliegt, bevor Bisognano–Wichmann-, Guido–Longo- oder Haag–Ruelle-Schritte beansprucht werden;
- $\beta$  auf dem aktiven Pfad nicht als stiller freier Endparameter verbleibt, sondern als modular abgeleitete oder Fixpunkt-gesteuerte Größe markiert ist;
- Split-Property nicht das Endziel bleibt, sondern eine explizite `NuclearitySeed/Nuclearity`-Schicht vorgesehen ist;
- `DHRsectorsSeed` und `FieldAlgebraReconstructionSeed` als spät-B/früh-E-Schicht für Ladung, Statistik und Eichgruppenrekonstruktion vorhanden sind;
- ein `BisognanoWichmannSeed` als explizite B->D-Brücke geplant ist;
- der alte `BoundaryMatrix`-Pfad als Bright-Spezialfall replayt werden kann, aber nirgends mehr als stilles Zentrum auftritt.

## 9.7 String-mathematische Ideen, die B vorbereiten muss

**Dokumentenebene:** Pfad-Ebene

Snowmass diskutiert Functorial Field Theory, conformal nets, homotopy AQFT und Defekte als reale QFT-Faces oder Erweiterungen. [4, pp. 9–12] Für CNNA bedeutet das in B ganz konkret: Defekte, Glueing, Boundary-State-Räume und Dualitätsdictionaries werden noch *nicht* de-seedet, aber die Datenstruktur muss so gelegt werden, dass sie später in B/E andocken können. Hier sind Atiyahs Gluing-Perspektive, Segals CFT-Begriff und die Defekt-/Boundary-Literatur aus der String-Mathematik konzeptionell einschlägig. [9][10][11][12]

## 10 Pillar C – offene Quantensysteme, Kanäle und Rückwirkung

**Dokumentebene:** Pfad-Ebene

### 10.1 Zielkern von C

**Dokumentebene:** Pfad-Ebene

Pillar C ist in CNNA nicht mehr nur der Ort, an dem eine bereits feststehende System-Umgebung- Interpretation mathematisch formalisiert wird. Der neue Kern ist: sektorielle Kanäle, Stinespring-artige Dilationslogik, Nicht-Markov-Strukturen, relative Entropie- und Flussschichten und daraus gewonnene Backreaction. Der Schritt von A/B nach C soll also nicht *Umgebung wird eliminiert*, sondern *Interface-vermittelte Wechselwirkung wird kanalisches expliziert* bedeuten.

### 10.2 Konkrete Vorgehensweise in C

**Dokumentebene:** Pfad-Ebene

- 1. Kanalgrundschrift von der Sektorgrammatik aus neu anschliessen.** Die mathematischen Unterlagen aus REALOQS/PillarC/OQS/Channel.lean und Stinespring.lean bleiben brauchbar. Inhaltlich müssen sie aber von einem binären SysEnv-Schema auf SectorChannels und SectorSysEnv umgestellt werden. Bright, Dark und Interface müssen als mögliche Quell-, Ziel- und Zwischenlagen der Kanäle explizit auftauchen.
- 2. Tail-Elimination als Kanal neu lesen.** TailElimAsChannel.lean ist kein Nebenthema, sondern die Scharnierdatei zwischen A und C. Sie muss zeigen, wie ein GeneralizedDtN-/MultiSchur-Schritt in eine effektive Kanalbeschreibung übersetzt wird. Erst dadurch wird der dunkle Sektor nicht mehr bloss *wegintegriert*, sondern als strukturierte Quelle effektiver Dynamik ernst genommen.
- 3. Endliche, semigruppenartige und nichtmarkovsche Lagen staffeln.** Finite, Lindblad, Semigroup und NonMarkov sollten nicht gleichzeitig als parallel konkurrierende Ontologien auftreten. Die Reihenfolge ist konkreter: zürst der allgemeinste sektorielle Kanaltrager, dann endliche/semigruppenartige Dynamik als kontrollierte Speziallage, anschliessend Nicht-Markov-Strukturen für genuin erinnerungsbehaftete Interface-Dynamik.
- 4. Relative Entropie und Flüsse an das Interface binden.** CNNA braucht für D und E keine abstrakte Entropierhetorik, sondern konkrete aus Kanälen und Zustandsrestriktionen ableitbare Flussgrößen. Deshalb sollte C früh eine RelativeEntropyFlow-Schicht erhalten, aus der später Backreaction, EffectiveLambda, HorizonLevel und entanglement-equilibrium-nahe Diagnosen gespeist werden.
- 5. DerivedSpacetime in C nur als Vorstufe lesen.** Das heute vorhandene DerivedSpacetime.lean in C zeigt bereits die Richtung: aus Einflussrelationen kann eine diskrete Raumzeitstruktur konstruiert werden. In CNNA darf dieses Objekt aber nicht Endpunkt von C bleiben. Es wird zum Vorbereitungsobjekt für D: aus Kanalfluss wird emergente Kausal- und Geometriestruktur. Genau darum ist das spätere Zielmodul CNNA/PillarD/DerivedSpacetime.lean und nicht mehr CNNA/PillarC/Integration/DerivedSpacetime.lean.

### 10.3 Abnahmekriterien für C

**Dokumentebene:** Pfad-Ebene

- sektorielle Kanäle sind definiert, bevor Lindblad-/Semigroup-Fenster de-seeden;
- Tail-Elimination ist als Kanal übersetzbar;
- relative Entropie- und Flussdaten sind an das Interface gebunden;
- Backreaction erscheint als abgeleitete, nicht als frei postulierte Zusatzdynamik;
- effektive Kanal- und Entwicklungslagen behalten auf dem aktiven Pfad eine explizit prüfbare Unitaritäts-/Positivitätskontrolle; semigruppenartige Spezialfenster bleiben de-seedete Sonderfälle.

### 10.4 Was in C nicht passieren darf

**Dokumentebene:** Pfad-Ebene

- der dunkle Sektor wird wieder nur als eliminierte Restumgebung gelesen;
- `DerivedSpacetime` wird als fertige Ontologie in C festgeschrieben;
- semigruppenartige Speziallagen werden als allgemeine Theorie verkauft;
- Entropie bleibt nur metaphorisch und wird nicht an konkrete Kanal- oder Zustandsdaten gebunden.

## 11 Pillar D – emergente Geometrie, lokale Kovarianz und Gravity-Routen

**Dokumentebene:** Pfad-Ebene

### 11.1 Zielkern von D

**Dokumentebene:** Pfad-Ebene

Pillar D ist der erste genuin neue CNNA-Pfeiler. Er darf nicht nur kinematische Etiketten auf ein schon vorhandenes Netz kleben, sondern muss den Kreis aus Kausalstruktur, Mass-/Skalenwahl, Materie-/Zustandssektor, Backreaction, Parameter-Closure und effektiver Einstein-/Friedmann-artiger IR-Dynamik schliessen. Die interne Spezifikation ist hier zweigleisig: primäre Jacobson-/Entanglement- Equilibrium-Route, sekundäre Route über induzierte Gravitation; asymptotic safety bleibt ein kompatibler, aber methodisch nachgeordneter Explikationsstrang. Gerade die Closure-Frage für  $b$ ,  $L_{\max}$ ,  $\beta$  und sekundäre Regularisierungsterme gehört deshalb nicht an den Rand, sondern in den D-Eingang selbst.[2, pp. 30–31, 34–35, 44–47]

**Kompositionsbild der Gravitation in CNNA:** Volle dynamische Raumzeit und volle dynamische Gravitation entstehen in CNNA nicht aus einem Einzelmechanismus, sondern aus der Komposition mehrerer abgeleiteter Stränge: Kausal-/Einflussstruktur, Mass-/Skalenfixierung, Zustands-/Materiesektor, Entropie-/Backreaction-Daten, sektorielle Eliminationsprozesse und — falls erfolgreich geschlossen — algebraisch-spektrogeometrische Zusatzstruktur. Jacobson, induced gravity und NCG sind deshalb nicht als monokausale Konkurrenzthesen, sondern als verschiedene Beitragsrouten zur selben emergenten Gravitationslesart zu führen.

### 11.2 Konkrete Vorgehensweise in D

**Dokumentebene:** Pfad-Ebene

**1. Aus Einflussrelation eine echte D-Sprache machen.** Nimm die in C vorbereiteten Einfluss- und Kanalrelationen und überführe sie in `CNNA/PillarD/DerivedSpacetime.lean`, `InterfaceCausality.lean` und `ComplementGeometry.lean`. Bereits vorhandene Kinematik-Dateien aus A — `DiscreteSpacetime`, `Causality`, `SpatialMetric`, `ProperTime*` — dürfen dabei nur als Seed-Material erscheinen. Ihre Begriffe werden erst de-seedet, wenn sie explizit aus `Bright/Dark/Interface-` und `Influence-`Daten rückgebunden sind.

**2. Lokal-kovariante Lesart vor geometrischer Spezialwahl herstellen.** Snowmass behandelt LCQFT als eigene, ernstzunehmende Face-Familie; zugleich zeigen BFV und spätere Arbeiten, dass lokale Kovarianz der natürliche Rahmen für QFT auf variablen Hintergründen ist.[4, pp. 6–9][6][18] Daher sollte D früh eine `LocalCovariance`-Schicht erhalten, bevor Minkowski-, Conformal- oder FLRW-Fenster überhaupt begonnen werden. Spezialfenster bleiben standardmäßig `Windows`.

**2a. Relative Cauchy Evolution und dynamical locality explizit als D-Dynamikschicht führen.** Lokale Kovarianz allein bleibt für CNNA zu statisch, sobald Hintergrundvariation, Stress-Tensor-Lesart und spätere Geometrie-/Backreaction-Aussagen ernst gemeint sind. Deshalb werden `CNNA/PillarD/RelativeCauchyEvolutionSeed.lean` und `CNNA/Gates/DynamicalLocality.lean` ausdrücklich zwischen `LocalCovariance` und späterer renormierter Observablen-Schicht verankert. So wird BFVs relative Cauchy evolution als operative Dynamik unter Hintergrundvariation und Fewster-Verchs dynamical locality als Konsistenztest zwischen kinematischer und dynamischer Lokalität in den Pfad eingebaut.[6][32]

**2b. Parameter-Closure als eigenen Zwischenstrang zwischen C und später D-Sprache führen.** Die Spezifikation behandelt Parameter-Closure nicht als Fussnote, sondern als eigene Schliessungsschwelle zwischen früher Dynamik und später Geometrie. Genau das muss auch die Roadmap sichtbar machen. Der Strang besitzt drei Ebenen: (i) ontische Mikrostrukturparameter wie  $b$ , die echte Verzweigung, Sektornichttrivialität und Mikrostruktur selektieren; (ii) dynamische Emergenzgrößen wie  $L_{\max}$ , die nur als Vorname einer späteren `HorizonLevel`- bzw. `BackreactionFixedPoint`-Lesart verbleiben dürfen; (iii) sekundäre numerische Regularisierer, die entweder ganz eliminiert oder als abgeleitete Restgrößen in eine `RegularizationClosure` überführt werden. Operativ heißt das: erst `BranchingWitness` mit einer sauberen Nichttrivialitätsbedingung wie `nontrivialSideBranching_iff_one_lt_b`; dann ein Selektorpfad `UVSpectralSelector->BranchingSelector->BackreactionSelector->selectedBranching`; danach `HorizonLevel`, `EffectiveLambda` und `BackreactionFixedPoint` als effektive IR-/Horizontschicht; schliesslich `RegularizationClosure` als Abschluss gegen freie numerische Restparameter. Erst wenn diese Closure-Schwelle erreicht ist, dürfen asymptotic-safety-Sprache, späte NCG-Interpretationen oder aggressive Window-Lesarten als mehr denn heuristische Diagnostik auftreten. Auf derselben Schiene muss auch  $\beta$  aus dem Status eines primitiven Gibbs-Inputs herausgezogen werden: entweder über den modularen/geometrischen Fluss oder als explizit markierte Fixpunktgröße innerhalb der Closure-Schicht.[2, pp. 44–47]

**3. Jacobson-/Entanglement-Route explizit am Interface aufhängen.** Jacobsons Idee, die Einstein-Gleichung als Zustandsgleichung aus `Horizon`-/Entropiebilanz zu lesen, ist für CNNA besonders passend, weil das Interface bereits der natürliche Sitz für Entropie- und Flussdiagnostik ist.[13] Konkret heißt das: aus `RelativeEntropyFlow`, `Backreaction` und lokalen Einflussmassen werden `DerivedStressTensorSeed`, `EntanglementEquilibriumSeed`, `HorizonThermodynamicSeed` und erst danach `EinsteinLimitSeed`. Dieser Strang ist der primäre D-Kern.

**3a. Hadamard-/mikrolokale Zustandskontrolle und lokale Wick-Produkte vor renormierter D-Sprache schliessen.** Sobald D über reine Kinematik hinaus auf `DerivedStressTensorSeed`, `EntanglementEquilibriumSeed` oder renormierte lokale Observablen zielt, reicht eine blossе GNS/KMS- oder Kausalitätsschicht nicht mehr aus. Deshalb werden `CNNA/PillarD/HadamardStateSeed.lean`, `MicrolocalSpectrumGate.lean` und `CNNA/PillarD/LocalWickPolynomialsSeed.lean` als echte D-Schicht eingeplant. Hadamard- bzw. mikrolokale Spektralkontrolle stabilisiert die zulässige Zustandsklasse, während lokale kovariante Wick- und Zeitordnungsprodukte spätere Stress-Tensor-, OPE- und Renormierungslesarten tragen.[33][34][35][36]

**3b. Eine explizite Connes-/NCG-Route als spätes D und frühe D $\leftrightarrow$ E-Brücke einfügen.** Nichtkommutative/spektrogeometrische Ansätze gehören in CNNA logisch nicht erst nach E, sondern setzen bereits in spätem D an: dort, wo geometrische Daten, Dirac-/Spektralstruktur und die Frage nach einer abgeleiteten Gravitationsdynamik selbst liegen. Deshalb werden `CNNA/PillarD/SpectralTripleSeed.lean`, `DiracOperatorSeed.lean`, `RealStructureSeed.lean` und `SpectralActionSeed.lean` als eigener D-Strang geführt. Er beginnt erst nach stabiler D-Kausalität, D-Skalenfixierung und B-Zustands-/Darstellungsschicht; sein E-naher Ausläufer ist nicht der Ursprung der Route, sondern höchstens eine spätere interne/finite Geometrie für Higgs-/Yukawa-/Flavor-Lesarten. Damit wird Connes/NCG als algebraisch-spektrogeometrische Kompositionsroute von D nach E eingeordnet, nicht als verkapptes E-Ersatzfundament.[19][20]

**3c. Induzierte Gravitation explizit als materiesektorielle Effektivroute lesen.** Um die Architektur nicht missverständlich zu machen, wird `InducedGravitySeed` sprachlich nicht

als Reserveideologie gegen Jacobson geführt, sondern als materiesektorielle Effektivroute, die neben der thermodynamischen Jacobson-Linie und der algebraisch-geometrischen NCG-Linie einen eigenen Beitrag zur emergenten Gravitationsdynamik liefert.

**4. Asymptotic-Safety-Kompatibilität als späte Closure-Diagnostik behandeln.** Die CNNA-Spezifikation verlangt ausdrücklich, dass asymptotic safety den derived-only Kern nicht früh dominiert. Das ist auch physikalisch sinnvoll: Reuter-artige Theorie-Raum-/Fixpunktlogik, Spectral-Dimension-Diagnostik und mode counting sind stark, aber sie müssen auf abgeleiteten Größen sitzen.[2, pp. 30–31][14][15][16] Also: zuerst `HorizonLevel`, `EffectiveLambda`, `BackreactionFixedPoint`, `RegularizationClosure` und ein explizites `gate_parameter_closure_complete`; danach erst `SpectralDimensionFlowSeed`, `CutoffModeCountingSeed`, `RunningBoundaryDataSeed`, `TheorySpaceSeed`. Die AS-Sprache bleibt erklärend und testend, bis Closure erreicht ist; dasselbe gilt für späte NCG-Deutungen und aggressive Windows, sofern sie primitive Parameter voraussetzen würden.

**5. Windows hart von Kernmodulen trennen.** `FLRWSectorWindow`, `ConformalWindow` und `MinkowskiWindow` dürfen nur dann auf den Kern rückwirken, wenn eine explizite Source-Map und ein de-seeding-Beschluss vorliegt. Das entspricht der internen Spezifikation und vermeidet, dass späte Regime heuristisch den ontischen Stamm überschreiben.[2, pp. 7–10, 31]

### 11.3 Abnahmekriterien für D

Dokumentebene: Pfad-Ebene

- `DerivedSpacetime`, `InterfaceCausality` und `LocalCovariance` sind aus C-Daten rückgebunden;
- `RelativeCauchyEvolutionSeed` und `DynamicalLocality-Gate` explizieren die D-Dynamik unter Hintergrundvariation;
- Hadamard-/mikrolokale Zustandskontrolle sowie `LocalWickPolynomialsSeed` stehen vor jeder renormierten Stress-Tensor- oder Einstein-Lesart;
- der Jacobson-Strang besitzt explizite Seeds für Stress-Tensor, Entanglement-Gleichgewicht und Einstein-Limit;
- ein expliziter Parameter-Closure-Strang schliesst  $b$ ,  $L_{\max}$ ,  $\beta$  und sekundäre Regularisierungsterme zu `selectedBranching`, `HorizonLevel/BackreactionFixedPoint` bzw. `RegularizationClosure`;
- die NCG-Route ist als spätes D und frühe D $\leftrightarrow$ E-Brücke mit `SpectralTripleSeed`, `DiracOperatorSeed`, `RealStructureSeed` und `SpectralActionSeed` sichtbar;
- induzierte Gravitation ist als getrennte späte materiesektorielle Effektivroute markiert;
- volle Gravitation wird explizit als Kompositionsresultat mehrerer Stränge gelesen und nicht monokausal behauptet;
- AS-Kompatibilität beginnt erst nach Closure-geeigneten Größen;
- Spezialfenster bleiben Windows.

### 11.4 Minimaler D-Dateibaum und explizite D/E-Gate-Reihenfolge

Dokumentebene: Pfad-Ebene

Analog zum E-Baum sollte auch D nicht nur begrifflich, sondern als minimaler Lean-Dateibaum sichtbar sein. Gerade dadurch wird klar, wo D als eigener Pfeiler endet und wo die Pfeilerübergreifende Brücke nach E wirklich beginnt.

Tabelle 28: Minimaler D-Dateibaum und explizite D/E-Gate-Reihenfolge – Teil 1

Zieldatei / Gate	Rolle	Stop-Regel
CNNA/PillarD/DerivedSpacetime.lean	erster expliziter Raumzeitträger aus Einfluss-, Closure- und Backreaktionsdaten	nach C-Flüssen und <code>gate_parameter_closure_complete</code> , vor <code>InterfaceCausality</code>
CNNA/PillarD/InterfaceCausality.lean	kausale/regionale Struktur auf dem Interface	nach <code>DerivedSpacetime</code> , vor <code>LocalCovariance</code> , <code>PTBridgeSeed</code> und späterer BW-Deutung
CNNA/PillarD/LocalCovariance.lean	funktoriale Lokalkovarianz des emergenten D-Layers	nach <code>InterfaceCausality</code> , vor <code>RelativeCauchyEvolutionSeed</code> und renormierter D-Sprache
CNNA/PillarD/RelativeCauchyEvolutionSeed.lean	Dynamik unter lokaler Hintergrundvariation	nach <code>LocalCovariance</code> , vor <code>DynamicalLocality</code>
CNNA/Gates/DynamicalLocality.lean	Konsistenz von kinematischer und dynamischer Lokalität	nach <code>RelativeCauchyEvolutionSeed</code> , vor <code>HadamardStateSeed</code> und <code>PTBridgeSeed</code>
CNNA/PillarD/HadamardStateSeed.lean	mikrolokal zulässige Zustandsklasse für emergente Hintergründe	nach <code>DerivedSpacetime</code> , vor <code>MicrolocalSpectrumGate</code> und <code>LocalWickPolynomialsSeed</code>
CNNA/PillarD/MicrolocalSpectrumGate.lean	prüft Hadamard-/mikrolokale Spektralkontrolle	nach <code>HadamardStateSeed</code> , vor <code>LocalWickPolynomialsSeed</code>
CNNA/PillarD/LocalWickPolynomialsSeed.lean	lokale Wick- und Zeitordnungsprodukte	nach <code>MicrolocalSpectrumGate</code> , vor <code>DerivedStressTensorSeed</code>
CNNA/PillarD/DerivedStressTensorSeed.lean	renormierte lokale Stress-Tensor-Lesart des D-Layers	nach <code>LocalWickPolynomialsSeed</code> , vor <code>EntanglementEquilibriumSeed</code> und <code>InducedGravitySeed</code>
CNNA/PillarD/EntanglementEquilibriumSeed.lean	Jacobson-/Entanglement-Vorzone	nach <code>DerivedStressTensorSeed</code> , vor <code>EinsteinsLimitSeed</code>
CNNA/PillarD/EinsteinsLimitSeed.lean	thermodynamische/entanglement-getriebene GR-Lesart	nach <code>EntanglementEquilibriumSeed</code> erst nach <code>InterfaceCausality</code> , <code>LocalCovariance</code> und geschlossener B-Darstellungslage
CNNA/PillarD/SpectralTripleSeed.lean	spektrogeometrischer Kern der NCG-Route	nach <code>EntanglementEquilibriumSeed</code> , vor <code>RealStructureSeed</code>
CNNA/PillarD/DiracOperatorSeed.lean	Dirac-Struktur des spektrogeometrischen D-Strangs	nach <code>SpectralTripleSeed</code> , vor <code>RealStructureSeed</code>
CNNA/PillarD/RealStructureSeed.lean	reeller/KO-Layer der NCG-Brücke	nach <code>DiracOperatorSeed</code> , vor <code>SpectralActionSeed</code>
CNNA/PillarD/SpectralActionSeed.lean	algebraisch-spektrogeometrische D↔E-Brücke	nach <code>RealStructureSeed</code> , vor <code>FiniteInternalGeometrySeed</code> in E
CNNA/PillarD/InducedGravitySeed.lean	materiesektorielle Effektivroute für Gravitation	nach <code>DerivedStressTensorSeed</code> und stabiler Zustands-/Materieschicht

Die D/E-Brücke ist dabei explizit, aber nicht monilinear. Eine minimale Gate-Reihenfolge sollte mindestens in folgenden Lagen sichtbar bleiben:

- D-Zulässigkeit:** `gate_parameter_closure_complete` und `gate_infinite_carrier_or_quasilocal_limit_defined` müssen stehen, bevor D als kontinuierliche/aqft-artige Lesart überhaupt belastbar wird.
- Geometrisch-dynamischer D-Kern:** `DerivedSpacetime`, `InterfaceCausality`, `LocalCovariance`, `RelativeCauchyEvolutionSeed`, `DynamicalLocality`, `HadamardStateSeed`, `MicrolocalSpectrumGate` und `LocalWickPolynomialsSeed` bilden die erste volle D-Schale.
- Symmetriebrücke B→D→E:** `BisognanoWichmannSeed`, `TimeReversalOnGNS`, `AntiunitarySeed`, `SpinRepresentationSeed`, `FermionicStatisticsSeed`, `StatisticsGate`, `SpinStatisticsTheorem`, `KramersGate` und `KramersTheorem` dürfen erst auf dieser D-zulässigen Lage aufsetzen.
- Charge-/CPT-Brücke:** `DHRsectorsSeed`, `FieldAlgebraReconstructionSeed`, `ChargeStatisticsCoherence`, `ChargeReconstructionTheorem`, `PTBridgeSeed`, `ChargeConjugationSeed`, `CPTGate` und `CPTTheorem` schliessen die Pfeilerübergreifende Ladungs- und CPT-Linie.
- Volle Materieorganisation:** Erst danach dürfen `GaugeSectorSeed`, `AnomalyInflowSeed`, `ChiralMatterSeed`, Higgs-/Flavor- und Matching-Fenster als späte E-Schicht einsetzen. Symmetrie- und Charge-Brücke können intern teilweise interleaven, aber Matter darf nicht vor beiden geschlossenen Brücken beginnen.

## 11.5 Theoretische Rahmung für D

Dokumentenebene: Pfad-Ebene

Dieser Pfeiler darf sich auf drei externe Ideenkomplexe stützen, ohne von ihnen kolonisiert zu werden: (1) lokale Kovarianz, relative Cauchy evolution und dynamical locality als QFT-Gesicht für variable Hintergründe,[6][32][7] (2) Hadamard-/mikrolokale Zustandskontrolle sowie lokale Wick-/Zeitordnungsprodukte für renormierte lokale Observablen auf gekrümmten Hintergründen,[33][34][35][36] (3) Jacobsons thermodynamische Route und spätere entanglement-orientierte GR-Lesarten,[13] (4) induced gravity als materiesektorielle Effektivroute,[17] (5) Connes/NCG und Spektralaktion als algebraisch-spektrogeometrische Zusatzroute mit Brückenwirkung nach E,[19][20] (6) asymptotic safety mitsamt Theorie-Raum-, Fixpunkt- und Spektraldimensionsdiagnostik.[14][15][16] CNNA übernimmt dabei die Organisationsideen, nicht deren Ontologie als schon entschiedenes Fundament.

## 12 Pillar E – später Gauge/Matter/SM-Strang aus der Stammtheorie

**Dokumentebene:** Pfad-Ebene

### 12.1 Zielkern von E

**Dokumentebene:** Pfad-Ebene

Pillar E ist in der internen Spezifikation ausdrücklich *spät*. Das Standardmodell darf nicht als fertiger Block auf die Stammtheorie geklebt werden, sondern muss in logisch getrennte Teilprobleme zerlegt und erst danach wieder zusammengesetzt werden: emergente Eichsektoren, chirale Materie, Anomalie-/Ladungsstruktur, Higgs/EWSB, Flavor/Yukawa, QCD-/Hadronfenster und schliesslich SM/SMEFT-Matching.[2, pp. 33–36] Kramers, CPT und Spin-Statistik sind dabei keine autonomen späten E-Zusätze, sondern Verbraucher der zuvor geschlossenen B-Kern- und B→D-Brückenstruktur; E darf sie nur noch nutzen, nicht ersetzen oder nachträglich improvisieren. Für Ladung, Eichgruppe und Teilchen/Antiteilchen-Sprache kommt zusätzlich eine observablenseitige DHR/DR-artige Schicht hinzu: `DHRSectorSeed` und `FieldAlgebraReconstructionSeed` müssen vor jeder belastbaren Gauge- und Charge-Lesart den Übergang von lokalisierter Sektorstruktur zu Feldalgebra und kompakter Eichgruppe tragen.[24][25] Genau das ist mit Snowmass und mit der mathematischen String-Literatur kompatibel: QFT hat viele Faces, und String-Mathematik ist hier nicht als neues Fundament relevant, sondern als Sprache für Defekte, Glueing, Dualitäten, Boundary States und Inflow-Mechanismen.[4, pp. 9–12][10][11][12] Nichtkommutative/spektrogeometrische Ansätze berühren E dabei zwar stark, aber nicht als deren eigentlichen Ursprung: Der Kern der NCG-Route sitzt in spätem D, während E nur den E-nahen Ausläufer einer möglichen internen/finiten Geometrie für Higgs-, Yukawa- und Flavor-Strukturen aufnimmt. Damit wird `FiniteInternalGeometrySeed` im Erfolgsfall als organisierende Fortsetzung, nicht als Ersatz von `GaugeSectorSeed`, `HiggsSectorSeed` oder `YukawaFlavorSeed` gelesen.[2, pp. 35–36][19][20]

### 12.2 Feynman-Audit für Matter: was schon vor E als Gate relevant ist

**Dokumentebene:** Pfad-Ebene

Feynmans Dirac-Gedankengang ist für CNNA nicht erst in einem späten Teilchenfenster relevant, sondern bereits als frühes Konsistenzschema für die Architektur von B, D und dem Übergang nach E. Der Kernpunkt seiner Vorlesung ist dreifach: (i) positive Energie und Relativität erzwingen Antiteilchen- und Paarerzeugungsstrukturen; (ii) die Rückwärtslesart von Dynamik und die Teilchen/Antiteilchen-Beziehung sind eng mit PT/C-Umdeutung verbunden; (iii) für Spin- $\frac{1}{2}$  hängt die Spin-Statistik-Verknüpfung an einem echten Vorzeichenmechanismus, nicht bloss an der Existenz irgendeiner involutiven Zeitumkehr.[44, pp. 2–11, 24–27, 33–38] Für CNNA folgt daraus: Matter darf nicht als thermisches oder gruppentheoretisches Label beginnen, sondern muss auf einer zuvor explizit geschlossenen Brückenschicht aus Spektrum, Zeitorientierung, Zeitumkehr, Ladung/Antiladung, Repräsentation und Statistik aufbauen. Das ist mit der internen Spezifikation kompatibel, die E als späten derived Strang behandelt und freie Gauge/Matter-Daten auf dem Kernpfad ausschliesst.[2, pp. 33–36]

**Feynman-basierte Designregel für CNNA:** *Time reversal alone is not matter.* Ein anti-lineares Involutionenobjekt auf Observablenebene ist nützlich, ersetzt aber weder Teilchen/Antiteilchen-Dictionaries noch Spinor-Repräsentationen, CAR/Fock-Strukturen, Austauschvorzeichen oder chirale Inflow-Gates.

Im inspectierten v0.0605-Code sind bereits mehrere Vorbausteine vorhanden. Die Module `REALOQS/PillarB/AQFT/TimeReversal.lean` und `REALOQS/PillarB/AQFT/Derived/BoundaryMatrixTimeReversal.lean` definieren eine anti-lineare, sternverträgliche und involutive

Zeitumkehr auf Observablenebene; konkret wird auf der Matrixalgebra  $A \mapsto \text{transpose}(A^*)$  und damit eintragweise komplexe Konjugation realisiert. Zusätzlich unterscheidet `REALOQS/PillarB/AQFT/TimeOrientation.lean` bereits zwischen einer `plus`- und einer `minus`-Orientierung der Dynamik, und die Intertwining-Sätze in der abgeleiteten Matrixschicht tauschen diese beiden Richtungen sauber gegeneinander aus.[1] Ebenso gibt es in `REALOQS/PillarB/AQFT/HaagKastler/SupportFull.lean` eine formale Vorwärtskegel-Struktur `forwardCone/ForwardCone`, und `REALOQS/PillarB/AQFT/Derived/TreeOfCliquesExtendedFromPillarA.lean` stellt darauf ein `SpectrumCondition`-Zeugnis für den strikten ToC-Pfad bereit.[1]

Gerade hier ist aber wissenschaftliche Disziplin wichtig. `REALOQS/PillarA/Core/BInterfaces/GlobalCone.lean` definiert `GlobalStateCone` als kompatible Familie lokaler Zustände unter Restriktion; das ist mathematisch sinnvoll, aber nicht dasselbe wie Feynmans Lichtkegel- bzw. spacelike-Argument. Dass beide Begriffe das Wort “cone tragen, darf in der Roadmap nicht zu einer semantischen Kurzschlusslesart führen. Die `derived-only`-Regel der CNNA-Spezifikation verbietet genau solche vorgezogenen physikalischen Identifikationen.[2, pp. 4–7][1]

Der heutige Matter-Vorbau in `REALOQS/PillarB/AQFT/Derived/EnergyFromBoundaryOp.lean` bleibt entsprechend bewusst schwach: `MatterTemplate` liefert eine endliche Konfigurations- und Gibbs-Struktur über einem aus dem Randoperator abgeleiteten Energiefunktional, aber weder Ladung, Antiteilchen, Spinor-Geometrie noch Austauschstatistik. Als thermischer Testträger ist das nützlich; als Materiesektor im Feynman-Sinn ist es unzureichend.[1]

### 12.3 Feynman-Audit-Matrix für die Planung von D/E

**Dokumentebene:** Pfad-Ebene

Die folgende Matrix legt fest, wie Feynmans Argument in die CNNA-Planung eingetragen wird. Sie ist bewusst kein physikalischer Endbeweis, sondern eine Gate-Liste gegen vorschnelle Bedeutungszuschreibungen.

**Tabelle 30: Feynman-Audit-Matrix für die Planung von D/E – Teil 1**

Feynman-relevante Forderung	Status in v0.0605	Konsequenz für die Roadmap
Positive Energie / Vorwärtskegel	formales Vorwärtskegel- und Spektrums-Gate vorhanden, aber nur als sehr schlanke Witness-Schicht	in D als <code>InterfaceCausality/DerivedSpacetime</code> vertiefen; in E nicht mit Teilchenphysik verwechseln
Zeitumkehr und Zeitorientierung	anti-lineare involutive Zeitumkehr auf Observablenebene sowie <code>plus/minus</code> -Dynamik vorhanden	als B/D-Vorbau behalten; vor E um Repräsentations- und Zustandswirkung erweitern
Teilchen/Antiteilchen-Dictionary	im aktuellen Archiv nicht formalisiert; kein eigener <code>ChargeConjugation</code> -Layer sichtbar	vor <code>GaugeSectorSeed</code> und <code>ChiralMatterSeed</code> explizite C-/PT-/CPT-Gates einführen
Spin- $\frac{1}{2}$ / $T^2 = -1$ -artige Struktur	im aktuellen Archiv nicht vorhanden; heutige Zeitumkehr ist involutiv auf Observablenebene	Matter darf nicht aus der bestehenden <code>TimeReversal</code> n-Schicht herausinterpretiert werden
Austauschvorzeichen / Bose-Fermi-Statistik Paarerzeugung / Inflow / spectator-Mechanismus	keine CAR/Fock-/Austauschschicht sichtbar physikalisch motivierend durch Feynman, aber im Code noch nicht als Materieformalismus vorhanden	eigener Statistikstrang vor vollwertiger chiraler Materie; kein stilles Importieren fermionischer Regeln in E als expliziter Inflow-/Boundary-/Defekt-Strang einplanen, nicht erst als späte Kommentierung

Die operative Schlussfolgerung lautet daher: Feynman ist für CNNA schon *vor* E nützlich, aber im Modus eines Architektur-Audits. Er liefert keine Abkürzung, mit der aus den vorhandenen Modulen bereits Matter hervorgezaubert werden darf. Stattdessen liefert er harte Vorbedingungen dafür, welche Brücken zwischen B, D und E geschlossen sein müssen, bevor von einem echten Materiesektor überhaupt gesprochen werden kann.[44, pp. 2–11, 33–38][2, pp. 33–36]

### 12.4 Konkrete Vorgehensweise in E

**Dokumentebene:** Pfad-Ebene

**1. Eichsektoren aus Defekten, Glueing und Interface-Struktur emergieren lassen.** Beginne E nicht mit einer postulierten SM-Eichgruppe. Schreibe stattdessen `GaugeSectorSeed`.

`lean` als derived Kandidat, der aus Interface-Netz, Defekten, Boundary-State-artigen Räumen und Glueing-Daten entsteht. Hier ist der string-mathematische Input klar: Defekte und Phasengrenzen tragen echte physikalische Information und können Symmetrien oder Dualitäten codieren.[12][11]

**1a. Vor jedem Gauge-Sektor eine echte PT/C/CPT-Brücke erzwingen.** Der bestehende v0.0605-Code besitzt bereits eine brauchbare Zeitumkehr- und Spektrumsprache, aber noch keine explizite Schicht für `Parity`, `ChargeConjugation` oder deren Wirkung auf Repräsentationen, Zustände und spätere Feld-/Teilchenobjekte. Weil Feynmans Antiteilchenlesart an genau dieser Rückwärts- und Umdeutungsstruktur hängt, müssen `PTBridgeSeed`, `ChargeConjugationSeed` und später ein klarer `CPTGate` vor jedem belastbaren Matter- oder Chiralitätsanspruch eingeplant werden.[44, pp. 10–11, 38–39][1]

**2. Chiralität als harten Engpass explizit isolieren.** Die Nielsen–Ninomiya-Obstruktion ist für jede diskrete/lattice-nahe Stammtheorie der zentrale Test. CNNA darf daher `ChiralMatterSeed` erst dann de-seeden, wenn Doubling-/Mirror-Probleme wirklich adressiert sind. Das spricht für einen expliziten Boundary-/Interface-/Inflow-Strang und gegen naive direkte Lattice-Identifikation.[21][22]

**2a. Spin-Statistik nicht als späte Fussnote, sondern als eigenes Gate behandeln.** Feynmans Argument zeigt, dass eine relativistische Materieschicht nicht schon durch die Existenz eines Hamiltonoperators oder einer KMS-Struktur gesichert ist. Für Spin- $\frac{1}{2}$  muss ein echter Vorzeichenmechanismus der Austausch- bzw. Loop-Amplituden verfügbar sein, und für Kramers-artige Zeitumkehr darf  $T^2$  nicht als freies Vorzeichen schweben, sondern muss auf einer graduierten Struktur im Stil  $T^2 = (-1)^F$  bzw. ihrer halbzahligen Spin-Spezialisierung ruhen. Deshalb braucht CNNA spätestens zwischen `ChiralMatterSeed` und `SMMatchingWindow` einen expliziten `SpinRepresentationSeed`, einen `FermionicStatisticsSeed` sowie ein `StatisticsGate`, das bosonische und fermionische Sektoren nicht bloss benennt, sondern formal auseinanderhält. Ohne diesen Schritt bleibt Matter im besten Fall thermische Klassifikation, nicht Teilchenmaterie.[44, pp. 33–37][41][42]

**3. Anomalien und Ladungsquantisierung als echte Gates implementieren.** `AnomalyCancellationGate.lean` und `ChargeQuantizationGate.lean` müssen vor jeder SM-kompatiblen Redeweise bestanden sein. Inflow- und String-Literatur zeigen, dass Anomalien keine späte kosmetische Korrektur sind, sondern Konsistenzbedingungen des ganzen Sektors.[22][23]

**4. Higgs/EWSB als eigenen Organisationssektor behandeln.** Aus einer emergenten Eichgruppe folgt noch kein Massenspektrum. Deshalb müssen `HiggsSectorSeed` und `EWSBSeed` als eigene derived Baustellen geführt werden. Hier sind nichtkommutative/spektrale Geometrie und String-inspirierte Boundary-/Defektbilder hilfreich, nicht weil CNNA sie direkt übernimmt, sondern weil sie zeigen, dass Symmetrie, Geometrie und Higgs-Daten algebraisch gemeinsam organisiert werden können.[19][20]

**5. Flavor/Yukawa bewusst vom Kern trennen.** `YukawaFlavorSeed` und `GenerationSeed` gehören in einen späten Parallelstrang. Die Familien- und Mischungsproblematik darf den Kern nicht blockieren; sie darf aber auch nicht still als freie Matrix oder freie Familienzahl eingeführt werden. Dieser Strang beginnt erst, wenn Gauge, Chiralität und Anomaliegates stabil sind.

**6. IR-Matching erst ganz am Schluss.** CNNA sollte nicht zu früh auf volle beobachtete Teilchenphänomenologie zielen. Zunächst braucht es ein kontrolliertes IR-Matching auf einen SM- oder SMEFT-artigen Sektor. Hier ist die moderne SMEFT-Literatur der richtige Vergleichsrahmen:

erst wenn der emergente IR-Sektor sauber definiert ist, lohnt sich die Sprache von Matching und Wilson-Koeffizienten.[45][46]

## 12.5 Minimaler Zieldateibaum für E und seine Vorzone

Dokumentebene: Pfad-Ebene

Die Spezifikation beschreibt E nicht nur begrifflich, sondern bereits als Modulraum. Die Roadmap übernimmt das jetzt explizit, damit E nicht hinter der Detailtiefe der Spezifikation zurückfällt. Der eigentliche Pfeiler E beginnt spät, besitzt aber eine *Vorzone* zwischen D und E, die bereits in AQFT/Gates verankert ist:

Tabelle 32: Minimaler Zieldateibaum für E und seine Vorzone – Teil 1

Zieldatei / Gate	Rolle	Stop-Regel
CNNA/AQFT/GradedStarAlgebra.lean	Z $\mathbb{Z}$ -graduierte Sternalgebra als algebraische Unterlage für fermionische Sektoren	vor jeder belastbaren Statistik- oder Kramers-Sprache
CNNA/AQFT/SeparatingProperty.lean	Rech-/Schlieder-artige Separationsschicht für relevante Komplementalgebren	nach ComplementGNS, vor TomitaTakesakiData
CNNA/AQFT/TomitaTakesakiData.lean	extrahiert Tomita-Operator, modulare Konjugation $J$ und modularen Operator $\Delta$	nach SeparatingProperty, vor ComplementKMS
CNNA/AQFT/ModularConjugation.lean	kontrolliert $JMJ = M'$ in Bright/Dark/Interface-Lesart	vor PT/CPT- und Kramers-Lesart
CNNA/AQFT/ComplementSpectrumCondition.lean	Vorwärtskegel-/Positivitäts- und Spektralkontrolle auf dem aktiven Komplementnetz	nach ComplementKMS, vor BisognanoWichmannSeed und HaagRuelleSeed
CNNA/AQFT/BisognanoWichmannSeed.lean	Brücke von modularer Automorphismengruppe zu abgeleiteten Boost-/Wedge-Strukturen	nach ComplementKMS und ComplementSpectrumCondition, vor PTBridgeSeed/CPTGate; Abgleich mit InterfaceCausality erst über späteres Gate
CNNA/AQFT/NuclearitySeed.lean	Phase-Space-/Dichteschicht jenseits blosser Split-Property, aber erst als Nach-KMS-Schicht	nach ComplementKMS, vor später Typ-III-/Charge-Lesart
CNNA/Gates/Nuclearity.lean	prüft die Nuclearity-/Phase-Space-Kontrolle des Netzes	nach NuclearitySeed, vor DHRsectorsSeed
CNNA/AQFT/DHRsectorsSeed.lean	lokalisierte Superselection-Sektoren des Observablenetzes	nach Nuclearity, vor FieldAlgebraReconstructionSeed und GaugeSectorSeed
CNNA/AQFT/FieldAlgebraReconstructionSeed.lean	DR-artige Rekonstruktion von Feldalgebra und kompakter Eichgruppe	nach DHRsectorsSeed, vor ChargeConjugationSeed und belastbarer Gauge-Sprache
CNNA/Gates/ChargeStatisticsCoherence.lean	prüft Kohärenz von Ladung, Statistik und rekonstruiertem Gauge-Layer	nach FieldAlgebraReconstructionSeed, vor GaugeSectorSeed
CNNA/AQFT/ChargeReconstructionTheorem.lean	expliziter DHR/DR-Abschluss für Feldalgebra-, kompakte Gauge- und Ladungsrekonstruktion	nach ChargeStatisticsCoherence, vor voller Gauge-/Matter-Lesart
CNNA/PillarD/RelativeCauchyEvolutionSeed.lean	dynamische Reaktion auf lokale Hintergrundvariation	nach LocalCovariance, vor DynamicalLocality und renormierter D-Sprache
CNNA/Gates/DynamicalLocality.lean	prüft die Konsistenz von kinematischer und dynamischer Lokalität	nach RelativeCauchyEvolutionSeed, vor späterer GR-/CPT-Brücke
CNNA/PillarD/HadamardStateSeed.lean	Hadamard-/mikrolokale Zustandsklasse auf emergenten Hintergründen	nach DerivedSpacetime, vor renormierten lokalen Observablen
CNNA/PillarD/MicrolocalSpectrumGate.lean	prüft mikrolokale Spektralkontrolle/Hadamard-Zulässigkeit	nach HadamardStateSeed, vor LocalWickPolynomialsSeed
CNNA/PillarD/LocalWickPolynomialsSeed.lean	lokale kovariante Wick- und Zeitordnungsprodukte	nach MicrolocalSpectrumGate, vor DerivedStressTensorSeed
CNNA/PillarD/SpectralTripleSeed.lean	spektrogeometrischer Kern für die NCG-Route in spätem D	erst nach stabiler InterfaceCausality, LocalCovariance und B-Zustands-/Darstellungsschicht
CNNA/PillarD/DiracOperatorSeed.lean	abgeleitete Dirac-Struktur für die spektrogeometrische D-Linie	nach SpectralTripleSeed, vor RealStructureSeed
CNNA/PillarD/RealStructureSeed.lean	realer Struktur-/KO-Layer der NCG-Brücke	nach DiracOperatorSeed, vor SpectralActionSeed
CNNA/PillarD/SpectralActionSeed.lean	algebraisch-spektrogeometrische Wirkungsrouten als $D \leftrightarrow E$ -Brücke	nach RealStructureSeed, vor möglicher interner/finiter Geometrie in E
CNNA/AQFT/TimeReversalOnGNS.lean	hebt observablenseitige Zeitumkehr auf den GNS-Raum	nach TomitaTakesakiData, vor jeder Kramers-, PT- oder Matter-Lesart
CNNA/AQFT/AntiunitarySeed.lean	formalisiert Antiunitarität der implementierten Zeitumkehr	direkt nach TimeReversalOnGNS
CNNA/AQFT/SpinRepresentationSeed.lean	halbzahlige Spin-/double-cover-Schicht	vor FermionicStatisticsSeed
CNNA/AQFT/FermionicStatisticsSeed.lean	trennt fermionische von bosonischen Sektoren	vor StatisticsGate und ChiralMatterSeed
CNNA/Gates/StatisticsGate.lean	prüft Spin-Statistik-Konsistenz auf graduiertem algebraischen Untergrund	nach FermionicStatisticsSeed, vor ChiralMatterSeed
CNNA/AQFT/SpinStatisticsTheorem.lean	explizites Endtheorem für die Spin-Statistik-Schicht der rekonstruierten CNNA-AQFT	nach StatisticsGate, vor voller chiraler Matter-Lesart
CNNA/PillarE/PTBridgeSeed.lean	koppelt aus D stammende Paritäts-/Raumspiegelungsdaten an die B-Zeitumkehrschicht	nach LocalCovariance/InterfaceCausality, vor CPTGate
CNNA/PillarE/ChargeConjugationSeed.lean	Teilchen/Antiteilchen- bzw. Ladungskonjugationslayer	vor CPTGate und jedem belastbaren Gauge-Sektor

Fortsetzung auf der nächsten Seite

Zieldatei / Gate	Rolle	Stop-Regel
CNNA/Gates/CPTGate.lean	pfeilerübergreifende B+D+E-Abnahme von PT/C/CPT auf Darstellungs- und Zustandsniveau	erst nach D-Kausalität/Geometrie und vor belastbarer Matter-Sprache
CNNA/AQFT/CPTTheorem.lean	explizites Endtheorem für die pfeilerübergreifende CPT-Schicht	nach CPTGate, vor voller Matter-Interpretation
CNNA/Gates/KramersGate.lean	trennt graduiertes $\Theta^2 = (-1)^F$ , Zeitumkehrkovarianz, Spektrum und Entartung	direkt vor KramersTheorem.lean
CNNA/AQFT/KramersTheorem.lean	explizites Endtheorem für Kramers-Entartung im graduierten Zeitumkehrsektor	nach KramersGate
CNNA/PillarE/GaugeSectorSeed.lean	emergente Eichsektoren aus Interface/Defekt/Glueing	erst nach PT/C/CPT-Vorzone
CNNA/PillarE/AnomalyInflowSeed.lean	explizite Interface-/Boundary-Inflow-Schicht für chirale Rand- und Materiesektoren	nach FieldAlgebraReconstructionSeed, ChargeStatisticsCoherence und GaugeSectorSeed, vor ChiralMatterSeed und AnomalyCancellationGate
CNNA/Gates/AnomalyInflow.lean	prüft, dass die chirale Anomalieelast wirklich durch Inflow getragen wird	nach AnomalyInflowSeed, vor ChiralMatterSeed
CNNA/PillarE/ChiralMatterSeed.lean	chirale Materie unter Doubling-/Inflow-Kontrolle	nur nach StatisticsGate, FieldAlgebraReconstructionSeed, AnomalyInflowSeed und AnomalyInflow
CNNA/PillarE/AnomalyCancellationGate.lean	Anomalieabnahme für belastbare Gauge-/Chiralitätsrede	vor ChargeQuantizationGate und SMMatchingWindow
CNNA/PillarE/ChargeQuantizationGate.lean	Ladungs- und Quantisierungsgate	vor Higgs-/Flavor- und SM-Matching
CNNA/PillarE/HiggsSectorSeed.lean	Higgs-/symmetry-breaking-Vorzone	nur nach stabiler Gauge- und Chiralitätsschicht
CNNA/PillarE/EWSBSeed.lean	elektroschwache Brechung als eigener abgeleiteter Strang	nicht vor HiggsSectorSeed
CNNA/PillarE/FiniteInternalGeometrySeed.lean	E-naher Ausläufer der NCG-Route für interne/finite Geometrie	nach SpectralActionSeed, vor belastbarer Higgs-/Yukawa-Organisation
CNNA/PillarE/YukawaFlavorSeed.lean	Flavor- und Yukawa-Strang	später Parallelstrang
CNNA/PillarE/GenerationSeed.lean	Generationen-/Mischungsstruktur	später Parallelstrang
CNNA/PillarE/QCDWindow.lean	spätes IR-Fenster für starke Wechselwirkung	bleibt Window
CNNA/PillarE/HadronWindow.lean	hadronisches Spezialfenster	bleibt Window
CNNA/PillarE/SMMatchingWindow.lean	IR-Matching auf SM-artigen Sektor	erst nach allen Konsistenzgates
CNNA/PillarE/SMEFTWindow.lean	spätes Effektivfeldtheorie-Fenster	bleibt Window

## 12.6 Abnahmekriterien für E

Dokumentenebene: Pfad-Ebene

- keine freie Wahl von Eichgruppe, Hyperladungen, Higgs-Dublett, Yukawa-Matrizen oder Familienzahl auf dem Kernpfad;
- GaugeSectorSeed, AnomalyInflowSeed, ChiralMatterSeed, PTBridgeSeed, ChargeConjugationSeed, SpinRepresentationSeed, FermionicStatisticsSeed, StatisticsGate, CPTGate, KramersGate, AnomalyCancellationGate und ChargeQuantizationGate sind explizit vorhanden;
- NuclearitySeed/Nuclearity, DHRsectorsSeed, FieldAlgebraReconstructionSeed, ChargeStatisticsCoherence und ChargeReconstructionTheorem schliessen den Weg von lokaler Observablenstruktur zu Ladung/Gauge rekonstruierbar;
- die B-Kernschicht liefert bereits GradedStarAlgebra, SeparatingProperty, TomitaTakesakiData, ModularConjugation und einen BisognanoWichmannSeed als belastbare Vorzone;
- CPTGate ist ausdrücklich als pfeilerübergreifendes Gate zwischen B, D und E benannt und wird nicht vor InterfaceCausality/LocalCovariance abgeschlossen; CPTTheorem.lean bleibt als eigenes Endtheorem sichtbar;
- KramersGate setzt graduiertes Zeitumkehrverhalten, GNS-Implementierung und Spektrumsdaten voraus und bleibt daher eine späte Vorbedingung vor KramersTheorem.lean;
- SpinStatisticsTheorem.lean, CPTTheorem.lean und KramersTheorem.lean sind als symmetrisch benannte Zieltheoreme des Symmetrie-/Statistik-Strangs explizit ausgewiesen;
- Inflow-/Boundary-Strategien für Chiralität sind Teil des Pfads und mit AnomalyInflowSeed/AnomalyInflow nicht mehr nur verbale Motivation;
- FiniteInternalGeometrySeed bleibt ein E-naher Ausläufer der D-seitigen NCG-Route und ersetzt weder GaugeSectorSeed noch HiggsSectorSeed oder YukawaFlavorSeed;

- die Roadmap verwechselt `GlobalStateCone` nicht mit physikalischem Lichtkegel und behandelt `TimeReversalOn` nicht als bereits hinreichende Materieschicht;
- `SMMatchingWindow` und `SMEFTWindow` sind echte *Windows*, keine voreilige Kernontologie.

## 12.7 Späte Horizontschicht für Teilchen-, Scattering- und IR-Physik

**Dokumentebene:** Pfad-Ebene

Zwischen lokaler Algebren-/Sektorstruktur und beobachtbarer QFT liegt eine eigene Teilchen-, Scattering- und Infrared-Schicht. Sie ist kein unmittelbarer Kernblocker mehr, soll in dieser Roadmap aber als physikalische Horizontergänzung explizit sichtbar bleiben. In der AQFT ist dies kein blosser Komfortaufsatz. Scattering ist der primäre experimentelle Zugang zur relativistischen QFT, und für langreichweitige Kräfte wie QED reicht die nackte DHR-Sprache nicht bis zur realistischen Ladungsphysik; hier braucht man eine charge-class-/IR-verfeinerte Lesart. Deshalb führt die Roadmap diese Schicht bewusst als späte, nicht-kernblockierende Horizontzone. [28][27][31]

**Tabelle 34: Späte Horizontschicht für Teilchen-, Scattering- und IR-Physik – Teil 1**

Zieldatei / Gate	Rolle	Stop-Regel
<code>CNNA/AQFT/ClusterDecompositionSeed.lean</code>	optionale diagnostische Vorzone für Vakuum-/Korrelationszerfall und spätere Teilcheninterpretation	nach <code>ComplementSpectrumCondition</code> , vor voller <code>HaagRuelleSeed</code> -Lesart
<code>CNNA/AQFT/HaagRuelleSeed.lean</code>	späte Scattering-Vorzone für asymptotische Mehrteilchen- und S-Matrix-Lesart	erst nach <code>ChargeReconstructionTheorem</code> , geeigneter Spektralkontrolle und bevorzugt expliziter Clusterdiagnostik
<code>CNNA/AQFT/ParticleInterpretationWindow.lean</code>	Fenster für asymptotische Teilchen-, Detektor- und Streuprozessinterpretation	nach <code>HaagRuelleSeed</code> ; bleibt <code>Window</code>
<code>CNNA/AQFT/InfraredChargeClassSeed.lean</code>	IR-/long-range-charge-Schicht jenseits strikt lokalisierter DHR-Sektoren	spät, nach <code>ChargeReconstructionTheorem</code> ; besonders relevant für QED-artige Langreichweite

## 12.8 Was die String-Mathematik hier genau liefert

**Dokumentebene:** Pfad-Ebene

Die String-Mathematik soll in CNNA *nicht* als neues Fundamentalpostulat auftreten. Ihre brauchbaren Ideen sind viel präziser:

- Glueing/Funktorialität statt monolithischer globaler Beschreibung;[9][10]
- Defekte, Phasengrenzen und Boundary States als Träger nichttrivialer Daten;[11][12]
- Inflow-Mechanismen für chirale/anomale Randsektoren;[22][23]
- algebraisch-geometrische Organisation von Gauge-/Higgs-Strukturen ohne simplem Import des Endresultats.[19][20]

Genau in diesem präzisen Sinn gehört die String-Mathematik in CNNA hinein.

## 13 Pfeilerübergreifende partielle Ordnung statt Phasenplan

**Dokumentebene:** Pfad-Ebene

Es wird explizit *kein* Phasenplan vorgegeben. Trotzdem braucht die Migration eine partielle Ordnung. Diese Ordnung ist nicht zeitlich, sondern logisch.

**Status von Spezifikation und Roadmap.** Die ursprüngliche CNNA-Spezifikation v0.1 bleibt motivisches und begriffliches Leitdokument, ist für die konkrete Migration aus dem Archivstand v0.0605 inzwischen aber in mehreren Punkten *unterdeterminiert*. Für Priorisierung, Stop-Regeln, Handoff-Reihenfolge, Notationspflicht und operative Initialisierung der Pfeiler ist daher die vorliegende Roadmap das primäre bindende Zwischenartefakt zwischen Altarchiv und Zielsystem. Wo Spezifikation und operative Migrationslogik auseinanderlaufen, steuert die partielle Ordnung dieser Roadmap die Priorisierung.

Die partielle Ordnung ist deshalb nicht nur eine statische Liste von Vorbedingungen. Sie wird aus dem aktiven Codepfad und den dazugehörigen Tabellen fortlaufend nachgeführt. Insbesondere werden immer dem aktuellen Stand des Codes entsprechend die Phasenpläne je Abschnitt der partiellen Ordnung erstellt. Wenn neue belastbare Codeeinsichten eine Notationsschicht, einen Handoff oder einen Gate-Übergang verschieben, dann ist dies nicht nachträgliche Dokumentkosmetik, sondern eine notwendige Aktualisierung des executable Dokuments und ggf. der zuvor ab selbst.

0. **Setup der Notationsschicht vor jeder Fachlogik:** Vor dem ersten fachlichen Schritt in einem Pfeiler sind `CNNA/Notation.lean` und die jeweils benötigten `CNNA/PillarX/Notation.lean`-Module anzulegen. Diese Initialisierung gehört zur Ordner- und Importstruktur selbst; sie darf nicht nachträglich als kosmetische Dokumentationspflege folgen. Lesbare `scoped notation`-Schichten müssen für alle tragenden Lean-Objekte des aktiven Pfads vorbereitet sein, bevor die erste fachliche Logik des betreffenden Pfeilers implementiert wird.
1. **A vor allem anderen:** Ohne `BranchPatch`, `ComplementSectorFamily`, `SectorSplit` und `GeneralizedDtN` darf weder ein dunkles Netz in B noch ein echter Kanal in C beginnen.
2. **Kein full-derived lokales QFT-Face ohne Komplementnetz:** Wer Typ-III-, CPT-, Spin-Statistik- oder Kramers-Aussagen auf dem aktiven Pfad will, muss die komplementbasierte Netz-, Zustands- und Modularschicht zürst schliessen; `Bright-only-Recovery` bleibt Regression, nicht Kernableitung.
3. **B-Algebra vor fermionischer Semantik:** Die AQFT-Grundschrift von B muss spätestens mit `GradedStarAlgebra` Z2-offen werden, bevor Spin-, Statistik- oder Kramers-Sprache belastbar ist.
4. **GradedStarAlgebra und SeparatingProperty nicht verwechseln:** Die Graduierung der Algebrenbasis und die Reeh-Schlieder-artige Separationslage des GNS-Vektors gehören zu verschiedenen logischen Achsen; weder darf die eine als importierte Folge der anderen behandelt werden noch umgekehrt.
5. **B-Netze vor B-Modularität/Thermik:** `ComplementLocalNet/InterfaceLocalNet` und Zustandsrestriktionen müssen vor `ComplementGNS`; `ComplementGNS` wiederum vor `SeparatingProperty`, `TomitaTakesakiData`, `ModularConjugation` und erst danach vor `ComplementKMS` stehen.
6. **Komplement-Spektrum vor BW/Scattering/Guido-Longo:** Eine aktive Spektralkontrolle auf dem Komplementnetz via `ComplementSpectrumCondition` bzw. `gate_spectrum_condition_on_complement_net` muss vor `BisognanoWichmannSeed`, `HaagRuelleSeed` und jeder belastbaren Spin-Statistik-Lesart geschlossen sein.

7. **Endlicher Bright-Patch ist nicht schon die unendliche AQFT-Lage:** `InfiniteCarrier`, quasilokale Vervollständigung oder eine äquivalente thermodynamische Limesmarkierung müssen explizit vorliegen, bevor unendliche Kommutanten-, Typ-III-, BW- oder Scattering-Lesarten belastbar beansprucht werden.
8. **B/C vor D:** D darf erst beginnen, wenn Einfluss-, Fluss- und Backreactiondaten in C sowie die relevanten Netz-, Zustands- und modularen Daten in B vorhanden sind.
9. **Volle Gravitation nur als Kompositionsresultat lesen:** Weder Jacobson noch induced gravity noch NCG dürfen als isolierte Monokausalroute missverstanden werden; die belastbare D-Lesart entsteht erst aus dem Zusammenspiel von Kausalstruktur, Skalenfixierung, Materie-/Zustandssektor, Entropie/Backreaction und ggf. spektrogeometrischer Zusatzstruktur.
10. **Parameter-Closure vor AS-, NCG- und später Fenster-Sprache:** `BranchingWitness/selectedBranching`, `HorizonLevel`, `EffectiveLambda`, `BackreactionFixedPoint` und `RegularizationClosure` müssen vor `SpectralDimensionFlow`, mode counting, `TheorySpaceSeed`, späten `SpectralActionSeed`-Deutungen und heuristischen Spezial-`Windows` als primitive Steuerdaten geschlossen sein.
11. **NCG erst nach stabiler D-Kausalität und vor später E-Organisation:** `SpectralTripleSeed`, `DiracOperatorSeed`, `RealStructureSeed` und `SpectralActionSeed` dürfen erst nach `InterfaceCausality`, `LocalCovariance` sowie stabiler B-Zustands-/Darstellungsschicht einsetzen, aber vor später Higgs-/Yukawa-/Flavor-Organisation in E.
12. **D-Kausalität/Geometrie vor Pfeilerübergreifender CPT-Brücke:** `PTBridgeSeed`, `ChargeConjugationSeed` und `CPTGate` dürfen erst nach `InterfaceCausality`, `LocalCovariance` und dem `BisognanoWichmannSeed` beansprucht werden.
13. **Keine belastbare Charge-/Gauge-Sprache vor DHR/DR-Rekonstruktion:** `GaugeSectorSeed`, `ChargeConjugationSeed` und spätere Teilchen-/Antiteilchen- und Eichgruppenrede dürfen erst nach `DHRSectorSeed`, `FieldAlgebraReconstructionSeed` und `ChargeStatisticsCoherence` auftreten.
14. **Anomaly-Inflow erst nach rekonstruierter Ladungs-/Feldschicht:** `AnomalyInflowSeed` und `AnomalyInflow` dürfen nicht bloss nach einer verbalen Gauge-Motivation erscheinen, sondern erst nach `FieldAlgebraReconstructionSeed`, `ChargeStatisticsCoherence` und belastbarer `GaugeSectorSeed`-Lesart; sie bleiben zugleich Vorbedingung von `ChiralMatterSeed`.
15. **Keine renormierte D-Sprache vor Relative-Cauchy-/Hadamard-/Wick-Schicht:** `DerivedStressTensorSeed`, `EntanglementEquilibriumSeed`, `EinsteinLimitSeed` und spätere lokale Observablen verlangen vorab `RelativeCauchyEvolutionSeed`, `DynamicalLocality`, `HadamardStateSeed`, `MicrolocalSpectrumGate` und `LocalWickPolynomialsSeed`.
16. **Symmetrie- und Statistikbrücke nach D, aber vor voller Materie:** Zwischen D und E müssen `TimeReversalOnGNS`, `AntiunitarySeed`, `SpinRepresentationSeed`, `FermionicStatisticsSeed`, `StatisticsGate`, `CPTGate` und `KramersGate` explizit geschlossen werden; `TimeReversalOn` allein reicht dafür nicht.
17. **Prototypisches Vorseilen nur als Seed/Scaffold:** Frühe Dummy-Schnittstellen für spätere Pfeiler sind erlaubt und erwünscht, dürfen aber niemals als bereits geschlossener Kern verkauft werden; sie prüfen Tragfähigkeit, nicht Vollständigkeit.
18. **Rückfluss von Erkenntnissen ist zulässig, aber gate-diszipliniert:** Wenn spätere Pfeiler frühere Typ- oder Exportverträge sprengen, müssen die betroffenen Module kontrolliert

wieder geöffnet, reklassifiziert und neu geschlossen werden, statt die Inkonsistenz in spätere Schichten zu verschieben.

- 19. **Recovery zuletzt:** Bright-Recovery und Replay alter Beispiele sind Regression, nicht Entwurf des Kerns.

### 13.1 Gate-Reife statt Parallelillusion

**Dokumentebene:** Pfad-Ebene

Die partielle Ordnung ist zugleich eine Schutzregel gegen Scheinparallelisierung. Solange ein Modul oder Exportvertrag nur **Seed**-Status hat, bleiben spätere Detailpläne bewusst weich. Erst ein explizit geschlossenes Gate verhärtet die nächste Planungsschicht. Umgekehrt darf späterer Architekturzwang frühere Gates wieder öffnen: wenn C oder D zeigt, dass ein A-seitiger Typvertrag zu eng war, wird nicht stillschweigend weitergebaut, sondern kontrolliert auf **Seed/Scaffold** zurückgestuft, nachgebessert und neu geschlossen. Genau diese Rückkopplung ist kein Fehler der Roadmap, sondern ihr Flexibilitätsmechanismus. Jeder aus dieser partiellen Ordnung später abgeleitete Meilenstein trägt daher ein eigenes *Notation Review* als Definition of Done: ein Gate gilt erst dann als belastbar, wenn die zugehörigen Kernobjekte auch über ihre zugeordnete Notationsschicht lesbar, registriert und fachlich reviewbar sind.

### 13.2 Operativer Unterpfad 6a–6f für den modularen B-Kern

**Dokumentebene:** Pfad-Ebene

Obwohl diese Roadmap keinen globalen Phasenplan verwendet, braucht der kritische Pfad im B-Kern einen expliziten Unterpfad. Die frühe algebraische Bedingung ist bereits in **GradedStarAlgebra** verankert; der eigentliche modulare Unterpfad lautet dann:

**Tabelle 36: Operativer Unterpfad 6a–6f für den modularen B-Kern – Teil 1**

Schritt	Kernobjekt	Funktion im kritischen Pfad
6a	ComplementGNS	erste komplementnetz-basierte Darstellungslage; noch nicht modular geschlossen
6b	SeparatingProperty	stärkt den GNS-Vektor für relevante Komplementalgebren von zyklisch zu zyklisch+separierend
6c	TomitaTakesakiData + ModularConjugation	extrahiert $S, J, \Delta$ und schliesst die modulare Darstellungsarchitektur $JMJ = M'$
6d	ComplementKMS	thermische Schicht auf bereits geschlossener modularer Darstellungsarchitektur
6e	ComplementSpectrumCondition + gate_spectrum_condition_on_complement_net	schliesst Spektral- und Vorwärtskegelkontrolle auf dem Komplementnetz vor jeder BW-, Guido-Longo- oder Haag-Ruelle-Lesart
6f	BisognanoWichmannSeed	explizite B→D-Brücke von modularen Automorphismen zu abgeleiteten Boost-/Wedge-Strukturen

Dieser Unterpfad ist nicht kosmetisch. Er verhindert genau den falschen Kurzschluss **ComplementGNS**->**ComplementKMS**->**spaetirgendwoCPT/Kramers**, der die modulare Mittelschicht unsichtbar machen würde.

### 13.3 Spät-B-/Früh-E-Ergänzung für Nuclearity, DHR/DR und physikalische Rekonstruktion

**Dokumentebene:** Pfad-Ebene

Auf stabiler modularer und HK-/Quasilokal-Grundlage schliesst der Pfad noch nicht automatisch zu einer physikalisch reicheren AQFT. Deshalb werden unmittelbar oberhalb des heutigen B-Kerns drei weitere Stufen explizit geführt: **Tabelle 38: Spät-B-/Früh-E-Ergänzung für Nuclearity, DHR/DR und physikalische Rekonstruktion – Teil 1**

Schritt	Kernobjekt	Funktion im erweiterten Pfad
6g	NuclearitySeed + Nuclearity	führt Phase-Space- und Freiheitsgradkontrolle jenseits blossen Split-Properties ein; bleibt aber eine Nach-KMS-Schicht
6h	DHRsectorsSeed	lokalisiert Superselection-Sektoren des Observablenetzes als Charge-/Statistikvorzone
6i	FieldAlgebraReconstructionSeed + ChargeStatisticsCoherence	rekonstruiert Feldalgebra/kompakte Eichgruppe und bindet Ladung an Statistik

Diese Zusatzstufen sind nicht späte Dekoration, sondern die sauberste Stelle, an der CNNA den Schritt von netz- und modularitätsbasierter Architektur zu rekonstruierbarer Charge-/Gauge-Sprache vollzieht.

### 13.4 Operativer Unterpfad PC-a–PC-d für Parameter-Closure zwischen C und D

**Dokumentebene:** Pfad-Ebene

Die Spezifikation behandelt Parameter-Closure als eigene Schliessungsschwelle des derived-only Pfads. Diese Roadmap führt daher zusätzlich zum modularen Unterpfad einen eigenen Closure-Unterpfad:

**Tabelle 40: Operativer Unterpfad PC-a–PC-d für Parameter-Closure zwischen C und D – Teil 1**

Schritt	Kernobjekt	Funktion im Closure-Pfad
PC-a	BranchingWitness + nontrivialSideBranching_iff_one_lt_b	macht echte Verzweigung und Nichttrivialität des dunklen Sektors explizit; $b > 1$ ist hier Marker, nicht Endparameter
PC-b	UVSpectralSelector + BranchingSelector + BackreactionSelector	führt spektrale, verzweigungs- und rückwirkungsbezogene Daten zu einer abgeleiteten selectedBranching-Lesart zusammen
PC-c	HorizonLevel + EffectiveLambda + BackreactionFixedPoint	ersetzt ein primitives $L_{\max}$ durch eine effektive IR-/Horizontgröße und bindet sie an Zustand, Entropie und Interface-Balance
PC-d	RegularizationClosure + gate_parameter_closure_complete	eliminiert sekundäre numerische Regularisierer oder degradiert sie zu abgeleiteten Restgrößen; markiert die Closure-Schwelle vor AS-/NCG-/Window-Sprache

Auf demselben Strang muss  $\beta$  den Status eines primitiven Gibbs-Inputs verlieren: entweder als modular/geometrisch normalisierte Größe oder als explizit markierte Fixpunktvariable innerhalb von BackreactionFixedPoint. Erst danach ist der Pfad gegen freie Restparameter weit genug abgeschlossen, um genuinely derived-only genannt zu werden.[2, pp. 44–47]

### 13.5 Drei harte Zusatzregeln für Abhängigkeiten und Gates

**Dokumentebene:** Pfad-Ebene

1. **Keine fermionische Semantik vor GradedStarAlgebra.** Statistik-, Spin- und Kramers-Sprache dürfen nicht auf einer rein ungraduierten AQFT-Basis freischwebend eingeführt werden.
2. **Kein full-derived QFT-Face ohne Komplementnetz.** Wer vN-/Typ-III-, CPT-, Spin-Statistik- oder Kramers-Lesarten auf dem aktiven Pfad beansprucht, muss zuvor die komplementbasierte Kette von ComplementLocalNet/InterfaceLocalNet über ComplementStateNet bis in die modulare Darstellungsarchitektur geschlossen haben; Bright-only-Recovery zählt dafür nicht.
3. **Keine Kramers-/Spin-Statistik-/CPT-Sprache vor der modularen B-Kette.** Vor jeder belastbaren PT/C/CPT-, Spin-Statistik- oder Kramers-Lesart muss die Kette ComplementGNS->SeparatingProperty->TomitaTakesakiData->ModularConjugation->ComplementKMS geschlossen sein.
4. **Kein derived-only  $\beta$  als stiller Alt-Export.** Solange  $\beta$  nur als gegebener Gibbs-Parameter aus dem Bright-Pfad mitläuft, bleibt ComplementKMS eine Scaffold-Lesart; der aktive Pfad braucht entweder modulare/geometrische Ableitung oder explizite Fixpunktmarkierung.
5. **Kein CPTGate vor D-Kausalität/Geometrie plus BisognanoWichmannSeed.** Das CPTGate ist das erste ausdrücklich Pfeilerübergreifende Gate zwischen B, D und E und darf logisch nicht vor InterfaceCausality/LocalCovariance abgeschlossen werden.

### 13.6 Neue Gates für den modularen B-Kern und die B->D-Brücke

**Dokumentebene:** Pfad-Ebene

Zusätzlich zu `gate_complement_GNS` und `gate_complement_KMS` sind auf dem aktiven Pfad mindestens die folgenden Gates explizit zu führen:

**Tabelle 42: Neue Gates für den modularen B-Kern und die B->D-Brücke – Teil 1**

Gate	Rolle
<code>gate_graded_statistics_defined</code>	prüft, dass fermionische/bosonische Sektorsprache auf einer graduierten AQFT-Basis sitzt
<code>gate_complement_separating</code>	stoppt den Pfad, falls die Separationsstufe zwischen <code>ComplementGNS</code> und Modularität fehlt
<code>gate_tomita_takesaki_defined</code> <code>gate_modular_conjugation_maps_</code>	sichert, dass $S$ , $J$ und $\Delta$ als CNNA-Objekte vorliegen
<code>codebright_to_complement</code> <code>gate_spectrum_condition_on_</code> <code>codecomplement_net</code>	operationalisiert die Bright/Dark/Interface-Lesart von $JMJ = M'$
<code>gate_modular_boost_</code> <code>consistent_with_</code> <code>interface_causality</code>	stoppt den Pfad, solange Vorwärtskegel- und Positivitätskontrolle nur aus Bright-Recovery stammt prüft die Bisognano–Wichmann-Brücke vor jeder belastbaren CPT-Lesart

## 13.7 Weitere Gates für physikalische Vollständigkeit jenseits des heutigen Kerns

Dokumentenebene: Pfad-Ebene

Zusätzlich zur modularen B-Kette müssen für eine physikalisch breiter belastbare Rekonstruktionsroute spätestens die folgenden Gates im Dokument benannt bleiben:

**Tabelle 44: Weitere Gates für physikalische Vollständigkeit jenseits des heutigen Kerns – Teil 1**

Gate	Rolle
<code>gate_nuclearity_defined</code> <code>gate_dhr_sectors_defined</code> <code>gate_field_algebra_</code> <code>reconstruction_</code> <code>defined</code>	prüft die Phase-Space-/Energielevel-Dichte jenseits blossen Split-Properties sichert lokalisierte Superselection-Sektoren des Observablenetzes stoppt den Pfad, falls Feldalgebra- und kompakte Eichgruppenrekonstruktion nur heuristisch bleiben
<code>gate_charge_statistics_</code> <code>codecoherence</code> <code>gate_relative_cauchy_evolution_</code> <code>codedefined</code>	prüft die Kohärenz von Ladung, Statistik und rekonstruiertem Gauge-Layer
<code>gate_dynamical_locality</code> <code>gate_microlocal_spectrum_defined</code> <code>gate_local_wick_polynomials_</code> <code>codedefined</code>	operationalisiert die lokale Dynamik unter Hintergrundvariation vergleicht kinematische und dynamische Lokalität im LCQFT-Strang sichert Hadamard-/mikrolokale Zulässigkeit für D-Zustände verlangt lokale kovariante Wick- und Zeitordnungsprodukte vor renormierter D-Sprache
<code>gate_parameter_closure_</code> <code>codecomplete</code>	markiert die derived-only-Schwelle gegen freie Restparameter in $b$ , $L_{\max}$ , $\beta$ und Regularisierern
<code>gate_horizon_level_replaces_</code> <code>codeprimitive_Lmax</code>	stoppt den Pfad, solange $L_{\max}$ noch als ontischer Aktivparameter statt als abgeleitete Horizontgröße auftritt
<code>gate_infinite_carrier_or_</code> <code>codequasilocal_limit_defined</code>	sichert den Übergang vom endlichen Bright-Approximanten zur unendlichen/quasilokalen AQFT-Lesart
<code>gate_cluster_decomposition_</code> <code>codechecked</code>	optionale Spätdiagnostik für Vakuum-/Korrelationszerfall vor voller Teilchen- und Scattering-Interpretation
<code>gate_prototype_interfaces_type_stable</code>	prüft, dass frühe B-/C-Verbraucherschnittstellen auf A-Basis ohne ad-hoc Typreparaturen compiliert werden
<code>gate_back_propagation_resolved</code>	markiert, dass aus späteren Pfeilern zurücklaufende Architekturzwänge sauber in frühere Source-Maps und Exporte eingearbeitet wurden

**Kurzform der Abhängigkeitsordnung:** A-Stammtheorie  $\prec$  B-Algebra/Netze + Infinite-Carrier/quasilokaler Limes  $\prec$  B-Modularität/GNS/KMS + Komplement-Spektralkontrolle + Nuclearity + C-Kanäle/Backreaction  $\prec$  Parameter-Closure/HorizonLevel/RegularizationClosure  $\prec$  D-Geometrie/Relative-Cauchy/Hadamard/Wick + späte NCG-Brücke  $\prec$  DHR/DR-Rekonstruktion + Pfeilerübergreifende CPT/Spin/Statistik-Brücke  $\prec$  E-Gauge/Matter/SM-Fenster, mit einem separaten späten Recovery-Strang für den alten Bright-Pfad.

## 14 Datei-/Kettenblätter für tragende Pfade

**Dokumentebene:** Datei-/Ketten-Ebene

Die folgenden Kettenblätter heben die wichtigsten tragenden Pfade der Roadmap in das regelwerksnahe Schema 1–6: Ziel, Definitionen, Voraussetzungen, Behauptung, Beweis der Kette, Zielabgleich. Sie ersetzen keine spätere Voll-Dokumentation atomarer Einzeltheoreme, machen aber bereits auf Kettenebene prüfbar, welche logischen Bögen dieses Dokument tatsächlich behauptet.

### 14.1 Kettenblatt des modularen B-Kerns

**Dokumentebene:** Datei-/Ketten-Ebene

**1. Ziel der Kette.** Die Kette soll zeigen, wie aus dem komplementären lokalen Netz über Zustand und Darstellung eine modulare Darstellungsarchitektur entsteht, die für Spektralkontrolle, Bisognano–Wichmann und spätere Statistik-/CPT-Lesarten tragfähig ist.

**2. Definitionen der Kette.** Zentrale Objekte sind `ComplementLocalNet`, `ComplementStateNet`, `ComplementGNS`, `SeparatingProperty`, `TomitaTakesakiData`, `ModularConjugation`, `ComplementKMS` und `ComplementSpectrumCondition`.

**3. Voraussetzungen der Kette.** Ein sektor-first Netzpfad aus A, kontrollierte Zustandsfamilien, positive Energie-/Unitaritätsannahmen soweit für Spektralkontrolle nötig, keine boundary-first Abkürzung.

**4. Behauptung der Kette.** Der operative Unterpfad 6a–6f stellt einen geschlossenen modularen B-Kern bereit, dessen Endpunkt nicht bloss Thermik, sondern eine B→D-tragfähige modulare Lage ist.

**5. Beweis der Kette.** `ComplementStateNet` liefert die Zustandslage, `ComplementGNS` die Darstellung, `SeparatingProperty` stärkt diese zu zyklisch+separierend, `TomitaTakesakiData` und `ModularConjugation` extrahieren die modulare Struktur, `ComplementKMS` schliesst die Thermikschicht, `ComplementSpectrumCondition` bindet diese an Vorwärtskegel-/Positivitätsdaten.

**6. Zielabgleich der Kette.** Die Roadmap behauptet damit ausdrücklich keinen fertigen BW- oder CPT-Beweis, wohl aber den geschlossenen Vorpfad, ohne den diese späteren Aussagen architektonisch nicht derived-only getragen wären.

### 14.2 Kettenblatt des Parameter-Closure-Strangs

**Dokumentebene:** Datei-/Ketten-Ebene

**1. Ziel der Kette.** Freie Restparameter sollen als offener Scaffold-Zustand sichtbar bleiben und auf dem aktiven Pfad in einen derived-only Closure-Strang überführt werden.

**2. Definitionen der Kette.** `BranchingWitness`, `UVSpectralSelector`, `BranchingSelector`, `BackreactionSelector`, `ParameterClosure`, `HorizonLevel`, `BackreactionFixedPoint`, `RegularizationClosure`.

**3. Voraussetzungen der Kette.** sektorisierte ToC-Daten, Spektraldiagnostik, Kanal- und Rückwirkungsdaten aus C, keine freie AS-/NCG-/Window-Sprache vor Closure.

**4. Behauptung der Kette.** Der Unterpfad PC-a–PC-d erzeugt eine Closure-Schwelle, nach der ontische, dynamische und numerische Restparameter nicht mehr unkontrolliert als freie Inputs fungieren.

**5. Beweis der Kette.** Aus `BranchingWitness` und Spektraldiagnostik entstehen Selektoren, diese speisen `ParameterClosure`; daraus folgen `HorizonLevel` und `BackreactionFixedPoint`; `RegularizationClosure` schliesst oder degradiert die numerische Stabilisierung zu abgeleiteten Restgrößen.

**6. Zielabgleich der Kette.** Das Dokument beansprucht hier keinen bereits formal geschlossenen Fixpunktsatz, wohl aber die zwingende Position dieser Kette als derived-only Schwelle vor später geometrischer und matter-seitiger Sprache.

### 14.3 Kettenblatt der D/E-Brücke

**Dokumentebene:** Datei-/Ketten-Ebene

**1. Ziel der Kette.** Die D/E-Brücke soll den Übergang von geometrisch-dynamischer Zulassung zu späteren Symmetrie-, Statistik-, Charge- und Matter-Schichten explizit staffeln.

**2. Definitionen der Kette.** `InterfaceCausality`, `LocalCovariance`, `RelativeCauchyEvolutionSeed`, `HadamardStateSeed`, `LocalWickPolynomialsSeed`, `DHRsectorsSeed`, `FieldAlgebraReconstructionSeed`, `CPTGate`, `StatisticsGate`, `KramersGate`.

**3. Voraussetzungen der Kette.** geschlossener B-Kern, D-Zulassung, Hadamard-/Wick-Schicht, DHR/DR-Vorzone, keine freie Matter-Etikettierung vor CPT-/Charge-/Statistikbrücke.

**4. Behauptung der Kette.** Zwischen D und E liegt eine explizite Gate-Reihenfolge: geometrische Zulassung, renormierte lokale Dynamik, Symmetrie-/Statistikbrücke, rekonstruierte Charge-/Gauge- Schicht und erst danach volle Matter-Organisation.

**5. Beweis der Kette.** D erzeugt erst die lokal-kovariante und renormierte Hintergrundlage; Hadamard/Wick stabilisieren die lokale QFT-Sprache; DHR/DR rekonstruieren Charge- und Feldalgebra; CPT-/Statistics-/Kramers-Gates binden dies an die späte Matter-Ebene.

**6. Zielabgleich der Kette.** Die Roadmap macht damit sichtbar, dass E nicht isoliert startet, sondern ein kontrollierter Endverbraucher der zuvor geschlossenen B-, C- und D-Schichten ist.

## 15 Operative Regeln für De-seeding, Regression und Benennung

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

### 15.1 De-seeding-Regel

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Ein *Seed*-, *Scaffold*- oder *Window*-Modul verliert sein Suffix nur dann, wenn (1) eine explizite Source-Map vorliegt, (2) der zentrale Definitionskern keine freien Cutoffs, Zustände, Dictionaries oder Beobachter mehr enthält, (3) das Modul ein Derived-only-Gate auf dem kritischen Pfad besteht und (4) für alle neu eingeführten tragenden Definitionen eine entsprechende, dokumentierte Notation in der jeweils einschlägigen `Notation.lean` existiert, die den mathematischen Standard der Spezifikation lesbar abbildet. Ohne diese Notationsbindung ist ein De-seeding abzubrechen. Diese Regel ist direkt aus der internen Spezifikation übernommen und hier um die operative Lesbarkeitsbedingung erweitert.[2, pp. 6–10]

### 15.2 Regression gegen den Altpfad

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Der alte *BoundaryMatrix*-/*Bright*-Pfad bleibt im Projekt, aber in neuer Rolle:

- als Regressionsorakel für *Bright-Recovery*,
- als Extraktionsquelle für wirklich allgemeine algebraische oder netztheoretische Lemmata,
- als Nachweis, dass CNNA den aktiven Altpfad später reproduzieren kann.

Nicht zulässig ist dagegen, die alte Datei- und Namensordnung schrittweise wieder zum Kern zu machen.

### 15.3 Benennungsregel und Immediate Readability Policy

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Unsuffixed Dateinamen sind nur für genuinely derived Kernobjekte reserviert. Alles andere bleibt bewusst als *Seed*, *Scaffold* oder *Window* markiert. Die Modulnamen müssen daher selbst schon Auskunft über den epistemischen Status geben; genau das fordert die interne CNNA-Spezifikation.[2, pp. 6–10]

Für Neuentwicklungen jenseits des Archivstands v0.0605 gilt zusätzlich eine *Immediate Readability Policy*: roher Lean-Name allein genügt für tragende Kernobjekte nicht. Neue Definitionen müssen entweder selbstsprechend benannt sein oder sofort über eine zugehörige Notationsschicht in `CNNA/Notation.lean` bzw. `CNNA/PillarX/Notation.lean` lesbar gemacht werden. Ein Modul, das nur über opake Projektnamen verständlich bleibt, gilt operativ als noch nicht reviewreif.

### 15.4 Konsistente Ableitung der Notation-Dateien aus der Roadmap

**Dokumentebene:** Projekt-Ebene und Meta-Audit-Ebene

Die globalen und pfeilerspezifischen `Notation.lean`-Dateien sind nicht als späte manuelle Schönheitskorrektur zu behandeln, sondern relational aus drei kanonischen Tabellenlagen der Roadmap abzuleiten:

1. der **Definitionskatalog** als Master-Menge aller tragenden Lean-Objekte;
2. das **Register** als normalisierte Oberflächen-, Symbol-, Status- und Meta-Lage;
3. der **Notations-Appendix** als Emissionsmatrix der tatsächlich exportierten Lesbarkeits-schicht.

Daraus folgen vier bindende Regeln:

- Register und Appendix dürfen kein Fachobjekt eintragen, das nicht im Definitionskatalog steht.
- Jede stabile sichtbare Oberflächenform muss auf genau einen kanonischen Lean-Namen oder auf einen explizit als *meta only* markierten Begriff zeigen.
- Jede exportierte Lesbarkeitsform braucht Zielmodul, Scope-/Namespace-Regel und konkrete Expansion auf den Lean-Term.
- Jede belastbare Codeeinsicht, die Benennung, Symbolik oder Scope eines tragenden Objekts ändert, muss zürst den Definitionskatalog, danach Register und Emissionsmatrix und erst dann die realen `Notation.lean`-Module aktualisieren.

Operativ heißt das:

- `CNNA/Notation.lean` wird ausschliesslich aus globalen Kernbegriffen und global zulässigen Alias-/Symbolformen gespeist;
- `CNNA/PillarA/Notation.lean` bis `CNNA/PillarE/Notation.lean` sowie `CNNA/AQFT/Notation.lean` und `CNNA/OQS/Notation.lean` werden aus den jeweiligen Pfeiler- oder fachbereichsspezifischen Tabellenzeilen emittiert;
- Reifegradmarker wie `Seed`, `Scaffold` und `Window` sowie Meta-Begriffe wie `TruthProtocol` und `ExpertReview` sind registerpflichtig, aber *nicht* automatisch notationsfähig;
- das Notationsreview ist fester Bestandteil jedes Meilensteins, der aus der partiellen Ordnung gewonnen wird.

Wo die technische Umsetzung automatisiert werden kann, ist eine generierte oder halbautomatisch abgegliche `Notation.lean`-Schicht aus den Roadmap-Tabellen ausdrücklich zu bevorzugen, solange die resultierende Lean-Notation kollisionsfrei, `scoped` kontrolliert und reviewfähig bleibt.

## 15.5 Repo-Infrastruktur, Importgraph und Rollback

**Dokumentenebene:** Projekt-Ebene und Meta-Audit-Ebene

Vor jeder fachlichen De-seeding-Welle gelten vier nichttheorematische Pflichtbedingungen:

1. **pinned Toolchain:** `lean-toolchain` und `lakefile.lean` dürfen während einer laufenden Migrationswelle kein bewegliches Ziel sein; jede harte Gate-Stufe arbeitet gegen eine eingefrorene Toolchain;
2. **Build-Aggregatoren:** `CNNA/BuildAll.lean` sowie alle `CNNA/PillarX/BuildAll.lean`-Dateien werden früh erzeugt und in CI bzw. lokaler Build-Automation aufgerufen;
3. **Importgraph-Audit:** für jeden größeren Modulblock ist vor dem De-seeding ein Importgraph mit Zyklentest Pflicht; gerade QuasiLocal-, Gates-, Meta- und Recovery-Blöcke dürfen keine stillen Rückimporte in den Kern erzeugen;
4. **Rollback-Artefakte:** jedes harte Build-Gate erzeugt einen eingefrorenen Rückfallpunkt (Archivsnapshot oder ähnlich strikte Versionsmarke), damit fehlgeschlagene De-seeding-Versuche nicht unbemerkt in spätere Pfeiler einsickern.

Diese Infrastrukturregeln sind kein Fremdthema, sondern unmittelbarer Teil der Migration. Gerade die im Archiv bereits vorhandenen Root-, Meta- und Example-Dateien zeigen, dass Import- und Regressionsstabilität nicht erst *nach* der Mathematik beginnen.

## 16 Schlussurteil

**Dokumentebene:** Projekt-Ebene

Die richtige Migration von REAL-OQS v0.0605 nach CNNA ist *keine* lineare Fortsetzung des alten BoundaryMatrix-Pfads. Sie ist eine begriffliche Neuorganisation des Projekts. Der portierbare Kern aus A–C wird beibehalten, aber semantisch gedreht:

- von Boundary-first zu Sector-first,
- von einzelmem Komplement zu organisierter Komplementfamilie,
- von eliminiertes Umgebung zu kanalisch explizierter Rückwirkung,
- von frühen Spezialfenstern zu später, streng kontrollierter Recovery,
- von einer einzigen QFT-Oberfläche zu mehreren aus derselben Stammtheorie emergierenden Faces.

Genau dadurch werden die Snowmass-Forderung nach einer offeneren, gesichtssensitiven QFT-Begriffslandschaft, die interne CNNA-Spezifikation und der tatsächliche Codebestand v0.0605 miteinander vertragen.[4][2][3] Die Modulmatrix der Appendices deckt den vollständigen Archivbestand ab; die verbleibenden offenen Punkte liegen damit nicht mehr auf der Ebene architektonischer Grundsatzfragen, sondern auf der Ebene letzter physikalischer Präzisierungen eines im Kern bereits geschlossenen Migrationspfads.

## A Erweiterte Modulmatrix für die Migration

Dokumentenebene: Projekt-Ebene und Pfad-Ebene

### A.1 Direkte oder nahezu direkte Ports

Dokumentenebene: Projekt-Ebene und Pfad-Ebene

Tabelle 46: Direkte oder nahezu direkte Ports – Teil 1

Altmodul	Ziel in CNNA	Disposition
REALOQS/PillarA/Core/Approximant.lean	CNNA/Core/Approximant.lean	nahezu direkt portieren; Namespace und Importkegel bereinigen
REALOQS/PillarA/Core/RegionCore.lean	CNNA/Core/RegionCore.lean	nahezu direkt portieren; nur Semantik auf Sektorbild umstellen
REALOQS/PillarA/Core/RegionNet.lean	CNNA/Core/RegionNet.lean	als Trager für Bright/Dark/Interface-Regionen erhalten
REALOQS/PillarA/Core/InfiniteCarrier.lean	CNNA/Core/InfiniteCarrier.lean	beibehalten; keine boundary-first
REALOQS/PillarA/Core/Selection.lean	CNNA/Core/Selection.lean	Zusatzsemantik beibehalten; Beobachterbegriff nicht frei einführen
REALOQS/PillarA/OQS/DtN.lean	CNNA/OQS/DtN.lean	als binärer Sonderfall erhalten; Ausgangspunkt für GeneralizedDtN
REALOQS/PillarA/OQS/DtNStabilized.lean	CNNA/OQS/DtNStabilized.lean	als technische Hilfsschicht erhalten, aber methodisch dezentralisieren
REALOQS/PillarB/AQFT/StarAlgebra.lean	CNNA/AQFT/StarAlgebra.lean	direkt portieren
REALOQS/PillarB/AQFT/State.lean	CNNA/AQFT/State.lean	direkt portieren
REALOQS/PillarB/AQFT/StateNet.lean	CNNA/AQFT/StateNet.lean	direkt portieren und später sektorisieren
REALOQS/PillarB/AQFT/LocalNet.lean	CNNA/AQFT/LocalNet.lean	direkt portieren und später sektorisieren
REALOQS/PillarB/AQFT/GNS.lean	CNNA/AQFT/GNS.lean	direkt portieren; ComplementGNS darauf aufbauen
REALOQS/PillarB/AQFT/KMS.lean	CNNA/AQFT/KMS.lean	direkt portieren; ComplementKMS darauf aufbauen
REALOQS/PillarB/AQFT/RelativeEntropy.lean	CNNA/AQFT/RelativeEntropy.lean	direkt portieren
REALOQS/PillarC/OQS/Channel.lean	CNNA/OQS/SectorChannels.lean	semantisch verbreitern von System-Umgebung auf Sektor-Kanäle
REALOQS/PillarC/OQS/Stinespring.lean	CNNA/OQS/Stinespring-kompatibleUnterlage	direkt als mathematische Grundschicht behalten

### A.2 Semantische Umschriften

Dokumentenebene: Projekt-Ebene und Pfad-Ebene

Tabelle 48: Semantische Umschriften – Teil 1

Altmodul	Ziel in CNNA	Disposition
REALOQS/PillarA/OQS/SysEnv.lean	CNNA/OQS/SectorSysEnv.lean	binären Split durch Mehrsektor-Split ersetzen
REALOQS/PillarA/Update/TailEliminationDtN.lean	CNNA/OQS/MultiSchur.lean+CNNA/Integration/TailElimAsChannel.lean	Tail-Elimination nicht nur eliminatorisch, sondern kanalisch lesen
REALOQS/PillarA/Core/TailEliminationCoherence.lean	CNNA/Core/ParameterClosure.lean+CNNA/OQS/RegularizationClosure.lean	Koherenz in Closure-/Backreaction-Sprache überführen
REALOQS/PillarC/Integration/DerivedSpacetime.lean	CNNA/PillarD/DerivedSpacetime.lean	von C-Endpunkt zu D-Vorbereitungsobjekt umdeuten
REALOQS/PillarC/Integration/Locality.lean	CNNA/PillarD/InterfaceCausality.lean+CNNA/PillarD/LocalCovariance.lean	lokale Einflussrelation in lokale Kovarianz überführen

### A.3 Bright-Spezialfall und Regression

Dokumentenebene: Projekt-Ebene und Pfad-Ebene

Tabelle 50: Bright-Spezialfall und Regression – Teil 1

Altmodul	Status	Folge
REALOQS/PillarA/Ideal/Adapter/TreeOfCliquesAQFTHandoff.lean	Bright-Spezialfall / Regressionsorakel	nicht Architekturzentrum, sondern Vergleichspfad
REALOQS/PillarB/AQFT/Derived/TreeOfCliquesFromPillarA.lean	Bright-Spezialfall / Regressionsorakel	nur replayen, nicht zentrieren
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixLocalNet.lean	Bright-Spezialfall unter CNNA/AQFT	alte Randnetz-Semantik als abgeleitete Restriktion rekonstruieren
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixStateNet.lean	Bright-Spezialfall unter CNNA/AQFT	nur als Restriktion auf Bright
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixGNS.lean	Bright-Spezialfall unter CNNA/AQFT	nur als Restriktion auf Bright
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixFullKMSFromL.lean	Bright-Spezialfall unter CNNA/AQFT	nur als Restriktion auf Bright
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixQuasiLocalClosure.lean	Bright-Spezialfall / Regression	keine architektonische Zentralstellung

---

REALOQS/PillarB/AQFT/Derived/BoundaryMatrixModular.lean	Bright-Spezialfall / Regression	modulare Daten später aus CNNA zuruckgewinnen
---	---------------------------------	---

---

Die explizite Negativregel für diesen Appendix lautet: `BoundaryMatrixModular.lean`, `BoundaryMatrixTimeReversal.lean` und verwandte BM-derived-Dateien bleiben Recovery- oder Extraktionsquellen. Sie dürfen die neue modulare B-Kernschicht aus `SeparatingProperty`, `TomitaTakesakiData`, `ModularConjugation` und `BisognanoWichmannSeed` weder ersetzen noch logisch abkürzen.

## B Vollständige Disposition des BoundaryMatrix-Derived-Blocks

**Dokumentenebene:** Projekt-Ebene und Datei-/Ketten-Ebene

Der komplette BoundaryMatrix\*-Block umfasst im Archiv 28 Dateien mit zusammen 7'972 LOC. Er wird in CNNA weder pauschal verworfen noch pauschal zum Kern erklärt. Die Roadmap trennt jetzt explizit zwischen Bright-Recovery, Extraktionsquellen für allgemeine Mathematik und Gate-/Auditquellen.

**Tabelle 52: Vollständige Disposition des BoundaryMatrix-Derived-Blocks – Teil 1**

Altmodul	LOC	Disposition in CNNA
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixAdditiveDynamics.lean	465	Bright-Recovery plus Extraktionsquelle für spätere Komplement-Additivitätsgates; nicht Kernaxiom
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixAdditiveDynamics		Entscheidungs- und Auditquelle für
Decision.lean	757	additiv vs. nichtadditiv; in
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixAdditiveProperties.lean	160	CNNA/Meta und CNNA/Gates auswerten
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixAdditiveStateNet.lean	241	Extraktionsquelle für additive Netz-/Zustandslemmata; erst nach stabilem Komplementnetz
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixAdditiveSubnet.lean	259	Bright-Recovery-Unterstrang; nie vor ComplementStateNet
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixAlgebra.lean	47	Bright-Recovery und Vergleichspfad für spätere Additivitätsabnahmen
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixCStar.lean	163	algebraische Extraktionsquelle; generische Lemmata nach CNNA/AQFT ziehen
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixDensityState.lean	590	C*-Extraktionsquelle; nicht stillschweigend verlieren
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixDynamicsFromL.lean	170	Gibbs-/Dichtezustandsquelle für Bright-Recovery und allgemeine Zustandslemmata
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixFullDynamics.lean	534	Bright-Dynamikquelle; nur als Recovery- oder Extraktionspfad
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixFullExp.lean	305	substanzielle Dynamikableitung; als Recovery- und Extraktionsquelle explizit behalten
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixFullKMSFromL.lean	807	Exponential-/Semigroup-Hilfsschicht; generische Teile extrahieren
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixGNS.lean	173	zentrales Bright-KMS-Recovery-Orakel; nicht vor ComplementKMS
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixGibbsState.lean	118	Bright-Recovery für GNS; kein Kernstartpunkt
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixKMSFromL.lean	236	thermische Zustandsquelle; generische Teile extrahieren
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixLocalNet.lean	594	thermische Extraktions- und Recovery-Quelle; späterer Bright-Vergleich
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixLocality.lean	112	Bright-Recovery-Lokalnetz; als Restriktion rekonstruieren, nicht als CNNA-Kern lesen
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixModular.lean	240	HK-/Locality-Extraktionsquelle für Recovery und Gates
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixNontriviality.lean	91	modulare Recovery-Schicht; erst nach GNS/KMS-Vergleich
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixObservableBasis.lean	114	Gate-/Regressionquelle für Nichttrivialität
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixPhases.lean	94	Extraktionsquelle für Observable-Basis-Sprache
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixPhysicalGibbs.lean	250	späte Phasen-/Diagnostikquelle; kein Kernmodul
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixQuasiLocalClosure.lean	225	thermische Extraktionsquelle mit physikalischer Lesart; nur nach Zustandskern
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixRichNetNonAdditive.lean	362	Bright-Recovery für QuasiLocal-Closure; nach ComplementClosure
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixSplitProperty.lean	343	explizite Nichtadditivitäts- und Auditquelle; in CNNA/Meta und Gates erhalten
REALOQS/PillarB/AQFT/Derived/BoundaryMatrixStateNet.lean	187	Split-Property-Quelle für Recovery und HK-Gates
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixTimeReversal.lean	235	Bright-Recovery-StateNet; keine Kernsemantik
REALOQS / PillarB / AQFT / Derived / BoundaryMatrixToCNontriviality.lean	100	Bright-Recovery plus Vorstufe für den Kramers-Strang; nicht als hinreichende Matter-Symmetrie lesen
		ToC-spezifische Gate-/Regressionsquelle

## C Weitere Pillar-A Module aus Core, Symmetry und Kinematics

**Dokumentenebene:** Projekt-Ebene und Datei-/Ketten-Ebene

Die folgende Tabelle legt für die bisher implizit gelassenen A-Module eine explizite Disposition fest. Sie verhindert, dass stille Importabhängigkeiten später als “unbeabsichtigter Legacy-Schleier” in CNNA wieder auftauchen.

**Tabelle 54: Bisher untererfasste A-Restmodule aus Core, Symmetry und Kinematics – Teil 1**

Altmodul	LOC	Disposition in CNNA
REALOQS/PillarA/Core/Gates.lean	182	als <code>CNNA/Gates/PillarAFoundation.lean</code> oder ähnliche Gate-Unterlage explizit portieren; kein stiller Root-Import
REALOQS/PillarA/Core/SymmetryCoverageGates.lean	119	Symmetrie-Auditschicht; in <code>CNNA/Gates/SymmetryCoverage.lean</code> überführen
REALOQS/PillarA/Core/ResistanceMetricCanonical.lean	226	D-Kinematik-/Geometrie-Seed; nicht verwerfen, aber vorerst nicht unsuffixed
REALOQS/PillarA/Core/Policy.lean	116	in <code>CNNA/ArchitecturePolicy.md</code> und Importregeln rückbinden; keine stille Nichtübernahme
REALOQS/PillarA/Core/MatrixNorms.lean	79	algebraische Hilfsschicht für DtN-/Operatorpfade; bei Bedarf direkt portieren
REALOQS/PillarA/Core/SubstrateClass.lean	68	ToC-/Substrat-Typeclass als explizite Kernunterlage erhalten
REALOQS/PillarA/Core/SymmetryAction.lean	106	Symmetrie-Seed für D/E; nicht vorzeitig physikalisch übersetzen
REALOQS/PillarA/Core/DynamicsEquivariant.lean	39	kleine, aber echte Äquivarianzquelle; in Symmetrie-/Dynamikgates einfließen lassen
REALOQS/PillarA/Core/InfiniteCarrierSymmetry.lean	53	Infinite-Carrier-Symmetrie als Hilfsschicht erhalten
REALOQS/PillarA/Core/NoetherHooks.lean	85	bewusst als Seed/Hook belassen, bis echte derived Symmetriestränge existieren
REALOQS/PillarA/Symmetry/ApproxIsometry.lean	117	D-/Kinematik-Seed; nicht still in allgemeine Geometrie umetikettieren
REALOQS/PillarA/Kinematics/DiscreteSpacetime.lean	55	D-Seed; erst aus Interface/Inflünce-Daten de-seeden
REALOQS/PillarA/Kinematics/Causality.lean	35	D-Seed; nicht vor InterfaceCausality verabsolutieren
REALOQS/PillarA/Kinematics/Worldline.lean	48	D-Seed für spätere Beobachter-/Trajektorienfenster
REALOQS/PillarA/Kinematics/ProperTime.lean	45	D-Seed; kein freier Zeitbegriff auf dem Kernpfad
REALOQS/PillarA/Kinematics/ProperTimeFromMetric.lean	89	D-Seed für spätere Metrikfenster; vorerst nicht Kern
REALOQS/PillarA/Kinematics/SpatialMetric.lean	23	kleine, aber semantisch starke D-Seed-Datei
REALOQS/PillarA/Kinematics/Clock.lean	79	Beobachter-/Takt-Seed; nur spät de-seeden
REALOQS/PillarA/Kinematics/FrameChange.lean	41	Frame-Seed; nicht vor lokaler Kovarianz
REALOQS/PillarA/Kinematics/SpacetimeRegions.lean	31	D-Seed für spätere Regionen-/Window-Sprache

## D Disposition des B-Gates- und Meta-Subsystems

Dokumentenebene: Projekt-Ebene und Datei-/Ketten-Ebene

### D.1 B-Gates als eigene Validierungsschicht

Dokumentenebene: Projekt-Ebene und Datei-/Ketten-Ebene

Tabelle 56: B-Gates als eigene Validierungsschicht – Teil 1

Altmodul oder Modulgruppe	LOC	Disposition in CNNA
REALOQS/PillarB/Gates/StarAlgebraBasic.lean + CStarNormBasic.lean	85	als CNNA/Gates/AQFT/StarAlgebraBasic.lean bzw. CStarNormBasic.lean portieren
REALOQS/PillarB/Gates/StateRestriction.lean + SplitProperty.lean	106	direkt in die neue Gate-Ebene übernehmen; für Komplement- und Bright-Recovery gleichermaßen relevant
REALOQS/PillarB/Gates/QuasiLocalUP.lean + QuasiLocalStateLift.lean	47	QuasiLocal-Gate-Unterlage; nicht durch blosses Umbenennen von QuasiLocal/Basic ersetzen
REALOQS/PillarB/Gates/LocalNetFunctoriality.lean	28	Vorstufe für spätere Glueing-/Functorialitätsgates
REALOQS/PillarB/Gates/HaagKastler.lean	79	Aggregator für HK-Gates; als CNNA/Gates/HaagKastler.lean beibehalten
REALOQS/PillarB/Gates/ HaagKastler/{Isotony, Locality,Spectrum, TimeSlice,Vacuum}.lean	230	direkte Gate-Portierung; zentrale Abnahmeschicht für das neue Komplementnetz, aber Spektrum/Vakuum müssen im CNNA-Kern neu als Komplementnetz-Lesart ausgefaltet werden
REALOQS/PillarB/Gates/ HaagKastler/{Additivity, BasisAdditivity, StrongAdditivity, Covariance}.lean	259	explizite HK- und Additivitätsgates; wichtig für Vergleich rich vs. additive net
REALOQS/PillarB/Gates/ \{ImportPolicy,Exports, ExampleFiniteClassical}.lean	73	Import-/Smoke-Test-Schicht; ausserhalb des Kerns, aber nicht unerwähnt

### D.2 Meta-Dateien als Regressions- und Auditstrang

Dokumentenebene: Projekt-Ebene und Meta-Audit-Ebene

Tabelle 58: Meta-Dateien als Regressions- und Auditstrang – Teil 1

Altmodul oder Modulgruppe	LOC	Disposition in CNNA
REALOQS/Meta/PillarB_ObservableDynamicsDecision.lean	257	als Legacy-Regressions- und Entscheidungsquelle für Additivität/Nichtadditivität explizit behalten
REALOQS/Meta/PillarB_RichAudit.lean + PillarB_StrictSurface.lean	99	Auditquelle für rich vs. strict surface; nicht 1:1 Kern, aber echte Regression
REALOQS/Meta/PillarB_ObservableAudit.lean + PillarB_ObservablePath.lean + PillarB_ObservableActivePath.lean	146	dokumentieren, welche Altpfade der Bright-Recovery-Strang später replaysen muss
REALOQS/Meta/PillarB_ObservableBasisFromA.lean + PillarB_ObservableClosure\*.lean	137	Closure-/Basis-Regressionsquellen; in CNNA/Meta/LegacyRegression/* übernehmen
REALOQS/Meta/PillarB_Closure\*.lean + PillarB_BQ8\Closed.lean + PillarB_BQ9\Closed.lean	184	dokumentieren, welche Closure-Zielbilder später wieder erreicht werden müssen
REALOQS/Meta/ PillarB_ActivePath.lean + PillarB_NoConfigHarnessIn CriticalPath.lean	83	direkte Vorlage für CNNA/Meta/CriticalPath.lean und NoLegacyInCriticalPath.lean
REALOQS/Meta/BuildAll.lean + restliche kleine Observable- Meta-Dateien	2	Smoke-Test-/Aggregatorrolle; mitportieren oder im Root-Meta abbilden

## E Quell-Ziel-Abgleich für Namens- und Topologiefragen

**Dokumentenebene:** Projekt-Ebene und Meta-Audit-Ebene

Tabelle 60: Quell-Ziel-Abgleich für Namens- und Topologiefragen – Teil 1

Quellmodul(e)	Ziel in CNNA	Bemerkung
REALOQS / PillarB / AQFT / QuasiLocal / \{Carrier,Structure,StarAlgebra\}.lean	CNNA/AQFT/QuasiLocal/Basic.lean	Basic ist neues CNNA-Aggregat, kein alter Dateiname
REALOQS/PillarB/AQFT/QuasiLocal/\{Completion,StateLift\}.lean	CNNA/AQFT/QuasiLocal/ComplementClosure.lean	ComplementClosure ist neue sektorisierte Lesart; bei Importdruck Hilfsunterdateien beibehalten
REALOQS/PillarB/AQFT/HaagKastler/Support.lean	CNNA/AQFT/HaagKastler/Basic.lean	Basic ist neues Zielminimum, nicht 1:1-Quellname
REALOQS/PillarB/AQFT/HaagKastler/SupportFull.lean	CNNA/AQFT/HaagKastler/Complement.lean	Voll-Support wird in die neue Komplementlesart überführt
REALOQS/PillarB/AQFT/HaagKastler/SplitProperty.lean	CNNA/AQFT/HaagKastler/SplitProperty.lean	1:1-Port mit neuer Importumgebung
REALOQS/PillarC/OQS/Finite/Kraus.lean	CNNA/OQS/Finite/Kraus.lean	explizit in den Zielbaum aufnehmen
REALOQS/PillarC/OQS/Finite/RectKraus.lean	CNNA/OQS/Finite/RectKraus.lean	explizit in den Zielbaum aufnehmen
REALOQS/PillarC/OQS/Finite.lean	CNNA/OQS/Finite.lean	bleibt Aggregator
REALOQS/PillarB/AQFT/Derived/LocalConfigNet.lean + StateFromConfigNet.lean + QuasiLocalStateFromConfigNet.lean	CNNA/AQFT/BrightRecovery/ConfigNet/*	nicht verschwinden lassen; später Regression und Konfigurationsvergleich
REALOQS/PillarB/AQFT/Derived/MatterFromPillarA.lean + EnergyFromBoundaryUp.lean + GibbsStateUnConfig.lean	CNNA/PillarE/LegacyMatterTests/*	explizit als schwacher thermischer Matter-Vorbau klassifizieren, nicht als E-Kern
REALOQS / PillarB / AQFT / Derived / TreeOfCliquesExtendedFromPillarA.lean	CNNA/AQFT/BrightRecovery/TreeOfCliquesExtendedFromPillarA.lean	Regressions- und Spektrumsquelle, insbesondere für Vorwärtskegel-/Matter-Audits

### E.1 Notationsmodule als Zielarchitektur

**Dokumentenebene:** Meta-Audit-Ebene

Für die reale Codebasis bedeutet die Notationspflicht eine eigene modulare Schicht. Minimal benötigt werden ein globales Modul `CNNA/Notation.lean` sowie bei Bedarf pillar- oder fachbereichsspezifische Ergänzungen wie `CNNA/PillarA/Notation.lean`, `CNNA/OQS/Notation.lean`, `CNNA/AQFT/Notation.lean`, `CNNA/PillarD/Notation.lean` und `CNNA/PillarE/Notation.lean`. Diese Module sind keine freie Stilzone, sondern direkte Emissionen der Roadmap-Tabellen. Deshalb gilt hier zusätzlich:

- der Definitionskatalog ist die einzige kanonische Objektquelle;
- das Register spaltet sich in Objekt-/Aliasregister, Symbolregister, Reifegradregister und Meta-Methodikregister;
- der Notations-Appendix enthält die tatsächlich exportierten Alias- und Notationsentscheidungen samt Zielmodul, Scope und Emissionsmuster;
- `selfspeaking`-Objekte bleiben bewusst ohne zusätzliche Notations-Emission, müssen aber im Definitionskatalog explizit als solche klassifiziert sein und können im Notations-Appendix als dokumentierte *Nicht-Emission* mitgeführt werden.

Jede spätere Tabellen- oder Registeränderung der Roadmap wirkt damit unmittelbar auf diese Notationsmodule zurück; die Notationsschicht ist also kein statischer Appendix, sondern ein direkt aus dem executable Dokument abgeleiteter Teil der Arbeitsbasis.

### E.2 Notations- und Alias-Appendix für den aktiven CNNA-Pfad

**Dokumentenebene:** Meta-Audit-Ebene

Die folgende Emissionsmatrix listet die im aktiven und spät aktivierten Pfad tatsächlich exportierten Lesbarkeitsformen sowie explizit dokumentierte `selfspeaking`-Nicht-Emissionen. Jede Zeile ist technisch determiniert: kanonischer Lean-Name, Emissionsart, Oberflächenform,

emittiertes Muster, Zielmodul, Scope und Kollisionsregel. Nicht exportierte, aber dennoch sichtbare Formen verbleiben ausschliesslich im Register; die einzigen Ausnahmen sind hier explizit markierte *Nicht-Emissionen* für tragende **selfspeaking**-Objekte. Die im Definitionskatalog geführte Primärform bleibt die dokumentarische Leitform; weicht die exportierte Kurzform davon ab, so ist sie als sekundäre Exportform sowohl hier als auch im Register I eigens auszuweisen.

Für jede emittierte Form sind Namespace, Aktivierungsregel, erforderlicher Kontext, Präzedenz bzw. Bindungsstärke, Kollisionsschutz, Beispielverwendung, Status und Review-Notiz tabellarisch festzuhalten. Keine dieser Informationen darf ausschliesslich aus dem Zielmodul oder aus dem Scope-Feld indirekt erschlossen werden.

**Tabelle 62: Notations- und Alias-Appendix – Teil 1**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel	erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
TreeOfCliques	text alias	<i>ToC</i>	scopedabbrevToC:=TreeOfCliques	CNNA/Notation.lean	CNNA	openscopedCNNA	global	n/a	<i>Tree of Cliques</i> bleibt Dokumentform; <i>ToC</i> ist der einzige Kurzalias	<i>ToC</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
BranchPatch	text alias	<i>Bright-Patch</i>	scopedabbrevBrightPatch:=BranchPatch	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A	n/a	kein zweiter Alias <i>Anker-Patch</i>	<i>Bright-Patch</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ComplementSectorFamily	scoped notation	<i>C</i>	scopednotation"C_comp"=>ComplementSectorFamily	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A–B	n/a	zusätzlicher Textalias <i>Komplementsektorfamilie</i> nur im Register	<i>C</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SectorSplit	scoped notation	$\Sigma_{BDI}$	scopednotation"SigmaBDI"=>SectorSplit	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A–B	n/a	keine konkurrierende Kurzform	$\Sigma_{BDI}$	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
GeneralizedDtN	scoped notation	$\Lambda_{gen}$	scopednotation"LambdaGen"=>GeneralizedDtN	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A–B	n/a	Textalias <i>generalisierter DtN-Kern</i> bleibt zulässig	$\Lambda_{gen}$	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SectorChannels	text alias	<i>SectCh</i>	scopedabbrevSectCh:=SectorChannels	CNNA/OQS/Notation.CNNA.OQS.lean		openscopedCNNA.OQS	Pillar C	n/a	Kurzalias für den kanalisierten OQS-Strang	<i>SectCh</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SectorSysEnv	text alias	<i>SysEnvSplit</i>	scopedabbrevSysEnvSplit:=SectorSysEnv	CNNA/OQS/Notation.CNNA.OQS.lean		openscopedCNNA.OQS	Pillar C	n/a	Kurzalias für den sektori-schen System/Environment-Split	<i>SysEnvSplit</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
RelativeEntropyFlow	text alias	<i>RelEntFlow</i>	scopedabbrevRelEntFlow:=RelativeEntropyFlow	CNNA/OQS/Notation.CNNA.OQS.lean		kontextgebunden; bei Export openscopedCNNA.OQS	Pillar C–D	n/a	Kurzalias für entropische Rückwirkungsflüsse	<i>RelEntFlow</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 63: Notations- und Alias-Appendix – Teil 2**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
ComplementLocalNet	scoped notation	$\mathcal{A}_{\text{comp}}$	scopednotation"A_comp"=>ComplementLocalNet	CNNA/AQFT/Notation.lean	CNNA.AQFT	openscopedCNNA.AQFT	n/a	Komplementnetz bleibt Registeralias	$\mathcal{A}_{\text{comp}}$	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
InterfaceLocalNet	scoped notation	$\mathcal{A}_{\text{int}}$	scopednotation"A_int"=>InterfaceLocalNet	CNNA/AQFT/Notation.lean	CNNA.AQFT	openscopedCNNA.AQFT	n/a	Interfacenetz bleibt Registeralias	$\mathcal{A}_{\text{int}}$	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ComplementStateNet	scoped notation	$\omega_{\text{comp}}$	scopednotation"omegaComp"=>ComplementStateNet	CNNA/AQFT/Notation.lean	CNNA.AQFT	openscopedCNNA.AQFT	n/a	kein Konflikt mit $\omega_{\beta}$	$\omega_{\text{comp}}$	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ComplementGNS	text alias	<i>CompGNS</i>	scopedabbrevCompGNS:=ComplementGNS	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	die Tupelform $(\pi_{\omega}, \mathcal{H}_{\omega}, \Omega_{\omega})$ bleibt Dokument-/Registerform	<i>CompGNS</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SeparatingProperty	text alias	<i>CycSep</i>	scopedabbrevCycSep:=SeparatingProperty	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	lesbare Kurzform nur für den Reeh-Schliederartigen Kontext	<i>CycSep</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
TomitaTakesakiData	text alias	<i>TTData</i>	scopedabbrevTTData:=TomitaTakesakiData	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	die Tupelform $(S, J, \Delta)$ bleibt Registerform	<i>TTData</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ComplementKMS	text alias	<i>CompKMS</i>	scopedabbrevCompKMS:=ComplementKMS	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	$\omega_{\beta}$ nur in explizitem KMS-Scope	<i>CompKMS</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ComplementSpectrumCondition	text alias	<i>CompSpectrum</i>	scopedabbrevCompSpectrum:=ComplementSpectrumCondition	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; B–D bei Export openscopedCNNA.AQFT	n/a	kein Kurzsymbol ohne Klartextkontext	<i>CompSpectrum</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 64: Notations- und Alias-Appendix – Teil 3**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel	erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
ParameterClosure	text alias	<i>ParamClosure</i>	scopedabbrevParamClosure:=ParameterClosure	CNNA/Notation.lean	CNNA	openscopedCNNA	global	n/a	bindet den derived-only-Closure-Strang sichtbar	<i>ParamClosure</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
EffectiveLambda	scoped notation	$\lambda_{\text{eff}}$	scopednotation"lambdaEff"=>EffectiveLambda	CNNA/PillarD/Notation.lean	CNNA.PillarD	openscopedCNNA.PillarD	C–D	n/a	keine Vermischung mit nacktem $\lambda$	$\lambda_{\text{eff}}$	planned active	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
RegularizationClosure	text alias	<i>RegClosure</i>	scopedabbrevRegClosure:=RegularizationClosure	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscopedCNNA.PillarD	C–D	n/a	lesbare Kurzform für Closure-Reviews	<i>RegClosure</i>	planned active	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
BranchingWitness	text alias	<i>BranchWitness</i>	scopedabbrevBranchWitness:=BranchingWitness	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscopedCNNA.PillarA	A–D	n/a	Witnessstyp nur in Branching-Kontexten	<i>BranchWitness</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
selectedBranching	text alias	<i>SelBranching</i>	scopedabbrevSelBranching:=selectedBranching	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscopedCNNA.PillarA	A–D	n/a	keine freie Umbenennung auf <i>chosen branching</i>	<i>SelBranching</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
UVSpectralSelector	text alias	<i>UVSel</i>	scopedabbrevUVSelector:=UVSpectralSelector	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscopedCNNA.PillarA	A–D	n/a	Selektor-Notation bleibt reviewpflichtig	<i>UVSel</i>	planned active	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
BranchingSelector	text alias	<i>BranchSel</i>	scopedabbrevBranchSelector:=BranchingSelector	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscopedCNNA.PillarA	A–D	n/a	Selektor-Notation bleibt reviewpflichtig	<i>BranchSel</i>	planned active	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
BackreactionSelector	text alias	<i>BackSel</i>	scopedabbrevBackSelector:=BackreactionSelector	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscopedCNNA.PillarD	C–D	n/a	Selektor-Notation bleibt reviewpflichtig	<i>BackSel</i>	planned active	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 65: Notations- und Alias-Appendix – Teil 4**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
HorizonLevel	scoped notation	$L_H$	scopednotation"L_H"=>HorizonLevel	CNNA/PillarD/Notation.lean	CNNA.PillarD	openscopedCNNA.PillarD C–D	n/a	ersetzt aktiv $L_{max}$ ; Legacyname nicht exportieren	$L_H$	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
BackreactionFixedPoint	text alias	<i>BackFix</i>	scopedabbrevBackFix:=BackreactionFixedPoint	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; C–D bei Export openscopedCNNA.PillarD	n/a	Kurzalias nur für die Fixpunktgleichung	<i>BackFix</i>	active core	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ComplementClosure	text alias	<i>CompClosure</i>	scopedabbrevCompClosure:=ComplementClosure	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT/C bei Export openscopedCNNA.AQFT	n/a	keine Symbolnotation ohne klaren Schließungskontext	<i>CompClosure</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ModularConjugation	text alias	<i>ModConj</i>	scopedabbrevModConj:=ModularConjugation	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	$J$ bleibt Symbolregistereintrag	<i>ModConj</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
BisognanoWichmannSeed	text alias	<i>BWSeed</i>	scopedabbrevBWSeed:=BisognanoWichmannSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	eindeutiger Kurzname für die BW-Brücke	<i>BWSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
NuclearitySeed	text alias	<i>NucSeed</i>	scopedabbrevNucSeed:=NuclearitySeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	keine Kurzform für Nuclearity selbst	<i>NucSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
DHRsectorsSeed	text alias	<i>DHRSeed</i>	scopedabbrevDHRSeed:=DHRsectorsSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	kurzer Reviewname für DHR-Vorzone	<i>DHRSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
FieldAlgebraReconstructionSeed	text alias	<i>FieldRecSeed</i>	scopedabbrevFieldRecSeed:=FieldAlgebraReconstructionSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export openscopedCNNA.AQFT	n/a	keine zweite Bedeutung für Rekonstruktionssatz	<i>FieldRecSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 66: Notations- und Alias-Appendix – Teil 5**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
ChargeStatisticCoherence	text alias	<i>ChargeStatCoh</i>	scopedabbrevChargeStatCoh:=ChargeStatisticsCoherence	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT/Gates bei Export opens copedCNNA.AQFT	n/a	Kurzalias nur für das Gate	<i>ChargeStatCoh</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ChargeReconstructionTheorem	text alias	<i>ChargeRec</i>	scopedabbrevChargeRec:=ChargeReconstructionTheorem	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	n/a	Kurzalias für Reviewtabellen	<i>ChargeRec</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
DerivedSpacetime	text alias	<i>DerSpacetime</i>	scopedabbrevDerSpacetime:=DerivedSpacetime	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	keine Konkurrenz zum Prosaausdruck <i>derived spacetime</i>	<i>DerSpacetime</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
InterfaceCausality	text alias	<i>IfcCaus</i>	scopedabbrevIfcCaus:=InterfaceCausality	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	Kurzalias nur für D-Reviews	<i>IfcCaus</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
LocalCovariance	text alias	<i>LocCov</i>	scopedabbrevLocCov:=LocalCovariance	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	keine Konkurrenz zu mathlib-Kontexten	<i>LocCov</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
HadamardStateSeed	text alias	<i>HadSeed</i>	scopedabbrevHadSeed:=HadamardStateSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	Kurzalias nur für Hadamard-Strang	<i>HadSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
MicrolocalSpectrumGate	text alias	<i>MicroSpec</i>	scopedabbrevMicroSpec:=MicrolocalSpectrumGate	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	Kurzalias nur für mikrolokales Gate	<i>MicroSpec</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
LocalWickPolynomialsSeed	text alias	<i>WickSeed</i>	scopedabbrevWickSeed:=LocalWickPolynomialsSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	keine Symbolnotation ohne lokalen Kovarianzkontext	<i>WickSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 67: Notations- und Alias-Appendix – Teil 6**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
EinsteinLimitSeed	text alias	<i>EinsteinSeed</i>	scopedabbrevEinsteinSeed:=EinsteinLimitSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	Kurzalias für späten GR-Limes	<i>EinsteinSeed</i>	planned active	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
TimeReversalOnGNS	text alias	<i>TRonGNS</i>	scopedabbrevTRonGNS:=TimeReversalOnGNS	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	n/a	∅ bleibt Symbolregister-eintrag	<i>TRonGNS</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SpinRepresentationSeed	text alias	<i>SpinSeed</i>	scopedabbrevSpinSeed:=SpinRepresentationSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	n/a	Kurzalias nur für den Spin-Strang	<i>SpinSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
StatisticsGate	text alias	<i>StatGate</i>	scopedabbrevStatGate:=StatisticsGate	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT/Gates bei Export opens copedCNNA.AQFT	n/a	Kurzalias für Statistikabnahme	<i>StatGate</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SpinStatisticsTheorem	text alias	<i>SpinStatThm</i>	scopedabbrevSpinStatThm:=SpinStatisticsTheorem	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	n/a	Kurzalias für Theoremtabellen	<i>SpinStatThm</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
PTBridgeSeed	text alias	<i>PTSeed</i>	scopedabbrevPTSeed:=PTBridgeSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; D–E bei Export opens copedCNNA.PillarE	n/a	keine Konkurrenz zu Paritäts-/Zeitumkehrsymbolen	<i>PTSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ChargeConjugationSeed	text alias	<i>CSeed</i>	scopedabbrevCSeed:=ChargeConjugationSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Kurzalias nur für Ladungskonjugations-Seed	<i>CSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
CPTGate	text alias	<i>CPTCheck</i>	scopedabbrevCPTCheck:=CPTGate	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; B–D–E bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Pfeilerübergreifendes Gate	<i>CPTCheck</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 68: Notations- und Alias-Appendix – Teil 7**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
CPTTheorem	text alias	<i>CPTThm</i>	scopedabbrevCPTThm:=CPTTheorem	CNNA/PillarE/Notation.lean	CNNA.AQFT	kontextgebunden; B–D–E bei Export opens copedCNNA.AQFT	n/a	Kurzalias für Theoremtabellen	<i>CPTThm</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
KramersGate	text alias	<i>KramersCheck</i>	scopedabbrevKramersCheck:=KramersGate	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; AQFT/E bei Export opens copedCNNA.PillarE	n/a	$\Theta^2 = (-1)^F$ bleibt Symbolregistereintrag	<i>KramersCheck</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
KramersTheorem	text alias	<i>KramersThm</i>	scopedabbrevKramersThm:=KramersTheorem	CNNA/PillarE/Notation.lean	CNNA.AQFT	kontextgebunden; AQFT/E bei Export opens copedCNNA.AQFT	n/a	Kurzalias für Theoremtabellen	<i>KramersThm</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
GaugeSectorSeed	text alias	<i>GaugeSeed</i>	scopedabbrevGaugeSeed:=GaugeSectorSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Kurzalias für emergenten Eichstrang	<i>GaugeSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
AnomalyInflowSeed	text alias	<i>InflowSeed</i>	scopedabbrevInflowSeed:=AnomalyInflowSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Inflow-Vorzone	<i>InflowSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
AnomalyCancellationGate	text alias	<i>AnomCancel</i>	scopedabbrevAnomCancel:=AnomalyCancellationGate	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E/Gates bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Anomalieabnahme	<i>AnomCancel</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
ChargeQuantizationGate	text alias	<i>ChargeQuant</i>	scopedabbrevChargeQuant:=ChargeQuantizationGate	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E/Gates bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Quantisierungsgate	<i>ChargeQuant</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
HiggsSectorSeed	text alias	<i>HiggsSeed</i>	scopedabbrevHiggsSeed:=HiggsSectorSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Higgs-Vorzone	<i>HiggsSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

**Tabelle 69: Notations- und Alias-Appendix – Teil 8**

Lean-Objekt	Emissionsart	Oberflächenform	Lean-Emission / Expansion	Zielmodul	Namespace	Aktivierungsregel erforderlicher Kontext	Präzedenz / Bindungsstärke	Kollisionsregel	Beispielverwendung	Status	Review-Notiz
EWSBSeed	text alias	<i>EWSBSeed</i>	scopedabbrevEWSBSeed:=EWSBSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	gleicher Kurzname wie Lean-Name, aber als gescripter Alias erfasst	<i>EWSBSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
YukawaFlavorSeed	text alias	<i>YukawaSeed</i>	scopedabbrevYukawaSeed:=YukawaFlavorSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Flavorstrang	<i>YukawaSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
GenerationSeed	text alias	<i>GenSeed</i>	scopedabbrevGenSeed:=GenerationSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Kurzalias für Generationsstrang	<i>GenSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SMMatchingWindow	text alias	<i>SMMatch</i>	scopedabbrevSMMatch:=SMMatchingWindow	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Window-Status bleibt erhalten	<i>SMMatch</i>	late window	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
SMEFTWindow	text alias	<i>SMEFTWin</i>	scopedabbrevSMEFTWin:=SMEFTWindow	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; E bei Export opens copedCNNA.PillarE	n/a	Window-Status bleibt erhalten	<i>SMEFTWin</i>	late window	Form exportiert; siehe Register I für Primär-/Sekundärtrennung
DiracOperatorSeed	text alias	<i>DiracSeed</i>	scopedabbrevDiracSeed:=DiracOperatorSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; D bei Export opens copedCNNA.PillarD	n/a	Kurzalias für die NCG-Route	<i>DiracSeed</i>	active extension	Form exportiert; siehe Register I für Primär-/Sekundärtrennung

## F Variablen-, Namens- und Notationsregister

**Dokumentebene:** Meta-Audit-Ebene

Dieses Register ist ab jetzt explizit in vier getrennte Tabellen aufgespalten. Damit werden Objekt- und Aliasformen, Symbole und Parameter, Reifegradmarker sowie Meta-Methodik sauber getrennt. Kein Registerteil darf neue Fachobjekte erzeugen; jede nicht-meta Form muss auf einen kanonischen Eintrag des Definitionskatalogs zeigen.

### F.1 Register I: Objekt- und Aliasregister

**Dokumentebene:** Meta-Audit-Ebene

Dieses Register führt alle stabilen sichtbaren Oberflächenformen, die über den Lean-Namen hinaus verwendet werden. Jede Form zeigt auf genau *einen* kanonischen Lean-Namen. Wird im Notations-Appendix eine von der Primärform abweichende Kurzform exportiert, so erscheinen Primärform und Exportform hier jeweils als getrennte Registereinträge.

Weicht die Exportform von der Primärform ab, so ist im Register I die Primärform mit Formrang *primär* und die exportierte Kurzform mit Formrang *sekundär exportiert* zu führen. Der Notations-Appendix enthält in diesem Fall ausschliesslich die exportierte Form; die Primärform bleibt dokumentarische Leitform, sofern sie nicht selbst emittiert wird.

**Tabelle 70: Register I: Objekt- und Aliasregister – Teil 1**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>Tree of Cliques</i>	text alias	primär	TreeOfCliques	CNNA	keine Export-aktivierung;	global	register only	Definitionskatalog / CNNA-Core	identischer Verweis auf TreeOfCliques
<i>ToC</i>	text alias	sekundär dokumentarisch	TreeOfCliques	CNNA	keine Export-aktivierung;	global	register only	Register-Alias des Stammobjekts	Kurzalias nur für TreeOfCliques
<i>Bright-Patch</i>	text alias	primär	BranchPatch	CNNA.PillarA	keine Export-aktivierung;	Pillar A	register only	Definitionskatalog / A-Core	expandiert auf BranchPatch
<i>C</i>	symbol	primär	ComplementSectorFamily	CNNA.PillarA	keine Export-aktivierung;	Pillar A–B	register only	Definitionskatalog / Complement	kontextgebundenes Symbol für ComplementSectorFamily
<i>Komplementsektorfamilie</i>	text alias	sekundär dokumentarisch	ComplementSectorFamily	CNNA.PillarA	keine Export-aktivierung;	Pillar A–B	register only	Definitionskatalog / Complement	identischer Verweis auf ComplementSectorFamily
$\Sigma_{BDI}$	symbol	primär	SectorSplit	CNNA.PillarA	keine Export-aktivierung;	Pillar A–B	register only	Definitionskatalog / SectorSplit	kontextgebundenes Symbol für SectorSplit
<i>Bright/Dark/Interface-Split</i>	text alias	sekundär dokumentarisch	SectorSplit	CNNA.PillarA	keine Export-aktivierung;	Pillar A–B	register only	Definitionskatalog / SectorSplit	identischer Verweis auf SectorSplit
$\Lambda_{gen}$	symbol	primär	GeneralizedDtN	CNNA.PillarA	keine Export-aktivierung;	Pillar A–B	register only	Definitionskatalog / OQS	kontextgebundenes Symbol für GeneralizedDtN
<i>generalisierter DtN-Kern</i>	text alias	sekundär dokumentarisch	GeneralizedDtN	CNNA.PillarA	keine Export-aktivierung;	Pillar A–B	register only	Definitionskatalog / OQS	identischer Verweis auf GeneralizedDtN
<i>Sektorkanäle</i>	text alias	primär	SectorChannels	CNNA.OQS	keine Export-aktivierung;	Pillar C	register only	Definitionskatalog / primäre Form	identischer Verweis auf SectorChannels
<i>SectCh</i>	export alias	sekundär exportiert	SectorChannels	CNNA.OQS	keine Export-aktivierung;	Pillar C	exported	Notations-Appendix / CNNA/OQS/Notation.lean	scopedabbrevSectCh:=SectorChannels
<i>Sektor-SysEnv-Split</i>	text alias	primär	SectorSysEnv	CNNA.OQS	keine Export-aktivierung;	Pillar C	register only	Definitionskatalog / primäre Form	identischer Verweis auf SectorSysEnv
<i>SysEnvSplit</i>	export alias	sekundär exportiert	SectorSysEnv	CNNA.OQS	keine Export-aktivierung;	Pillar C	exported	Notations-Appendix / CNNA/OQS/Notation.lean	scopedabbrevSysEnvSplit:=SectorSysEnv
<i>relativer Entropiefluss</i>	text alias	primär	RelativeEntropyFlow	CNNA.OQS	keine Export-aktivierung;	Pillar C–D	register only	Definitionskatalog / primäre Form	identischer Verweis auf RelativeEntropyFlow

**Tabelle 71: Register I: Objekt- und Aliasregister – Teil 2**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>RelEntFlow</i>	export alias	sekundär exportiert	RelativeEntropyFlow	CNNA.OQS	kontextgebunden; Pillar C-D bei Export opens		exported	Notations-Appendix / CNNA/OQS/Notation.lean	scopedabbrevRelEntFlow:=RelativeEntropyFlow
<i>Komplementschließung</i>	text alias	primär	ComplementClosure	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT/C	register only	Definitionskatalog / Closure	identischer Verweis auf ComplementClosure
<i>CompClosure</i>	export alias	sekundär exportiert	ComplementClosure	CNNA.AQFT	kontextgebunden; AQFT/C bei Export opens		exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevCompClosure:=ComplementClosure
<i>Parameter-Closure</i>	text alias	primär	ParameterClosure	CNNA	keine Export-aktivierung; Registerform	global	register only	Definitionskatalog / Closure	identischer Verweis auf ParameterClosure
<i>ParamClosure</i>	export alias	sekundär exportiert	ParameterClosure	CNNA	openscopedCNNA	global	exported	Notations-Appendix / CNNA/Notation.lean	scopedabbrevParamClosure:=ParameterClosure
$\lambda_{\text{eff}}$	symbol	primär	EffectiveLambda	CNNA.PillarD	keine Export-aktivierung; Registerform	C-D	register only	Definitionskatalog / Closure	kontextgebundenes Symbol für EffectiveLambda
<i>Regularisierungsclosure</i>	text alias	primär	RegularizationClosure	CNNA.PillarD	keine Export-aktivierung; Registerform	C-D	register only	Definitionskatalog / Closure	identischer Verweis auf RegularizationClosure
<i>RegClosure</i>	export alias	sekundär exportiert	RegularizationClosure	CNNA.PillarD	kontextgebunden; C-D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevRegClosure:=RegularizationClosure
<i>Branching-Witness</i>	text alias	primär	BranchingWitness	CNNA.PillarA	keine Export-aktivierung; Registerform	A-D	register only	Definitionskatalog / Closure	identischer Verweis auf BranchingWitness
<i>BranchWit</i>	export alias	sekundär exportiert	BranchingWitness	CNNA.PillarA	kontextgebunden; A-D bei Export opens		exported	Notations-Appendix / CNNA/PillarA/Notation.lean	scopedabbrevBranchWit:=BranchingWitness
<i>selected branching</i>	text alias	primär	selectedBranching	CNNA.PillarA	keine Export-aktivierung; Registerform	A-D	register only	Definitionskatalog / Closure	identischer Verweis auf selectedBranching
<i>SelBranch</i>	export alias	sekundär exportiert	selectedBranching	CNNA.PillarA	kontextgebunden; A-D bei Export opens		exported	Notations-Appendix / CNNA/PillarA/Notation.lean	scopedabbrevSelBranch:=selectedBranching
<i>UV-Spektralselektor</i>	text alias	primär	UVSpectralSelector	CNNA.PillarA	keine Export-aktivierung; Registerform	A-D	register only	Definitionskatalog / Selektoren	identischer Verweis auf UVSpectralSelector
<i>UVSel</i>	export alias	sekundär exportiert	UVSpectralSelector	CNNA.PillarA	kontextgebunden; A-D bei Export opens		exported	Notations-Appendix / CNNA/PillarA/Notation.lean	scopedabbrevUVSel:=UVSpectralSelector

**Tabelle 72: Register I: Objekt- und Aliasregister – Teil 3**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>Branching-Selektor</i>	text alias	primär	BranchingSelector	CNNA.PillarA	keine Export-aktivierung; Registerform	A–D	register only	Definitionskatalog / Selektoren	identischer Verweis auf BranchingSelector
<i>BranchSel</i>	export alias	sekundär exportiert	BranchingSelector	CNNA.PillarA	kontextgebunden; keine Export-aktivierung; Registerform	A–D	exported	Notations-Appendix / CNNA/PillarA/Notation.lean	scopedabbrevBranchSel:=BranchingSelector
<i>Backreaction-Selektor</i>	text alias	primär	BackreactionSelector	CNNA.PillarD	keine Export-aktivierung; Registerform	C–D	register only	Definitionskatalog / Selektoren	identischer Verweis auf BackreactionSelector
<i>BackSel</i>	export alias	sekundär exportiert	BackreactionSelector	CNNA.PillarD	kontextgebunden; keine Export-aktivierung; Registerform	C–D	exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevBackSel:=BackreactionSelector
$L_H$	symbol	primär	HorizonLevel	CNNA.PillarD	keine Export-aktivierung; Registerform	C–D	register only	Definitionskatalog / Pillar D	kontextgebundenes Symbol für HorizonLevel
<i>Backreaction-Fixpunkt</i>	text alias	primär	BackreactionFixedPoint	CNNA.PillarD	keine Export-aktivierung; Registerform	C–D	register only	Definitionskatalog / Pillar D	identischer Verweis auf BackreactionFixedPoint
<i>BackFix</i>	export alias	sekundär exportiert	BackreactionFixedPoint	CNNA.PillarD	kontextgebunden; keine Export-aktivierung; Registerform	C–D	exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevBackFix:=BackreactionFixedPoint
$\mathcal{A}_{\text{comp}}$	symbol	primär	ComplementLocalNet	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	kontextgebundenes Symbol für ComplementLocalNet
<i>Komplementnetz</i>	text alias	sekundär dokumentarisch	ComplementLocalNet	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf ComplementLocalNet
$\mathcal{A}_{\text{int}}$	symbol	primär	InterfaceLocalNet	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	kontextgebundenes Symbol für InterfaceLocalNet
<i>Interfacenetz</i>	text alias	sekundär dokumentarisch	InterfaceLocalNet	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf InterfaceLocalNet
$\omega_{\text{comp}}$	symbol	primär	ComplementStateNet	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	kontextgebundenes Symbol für ComplementStateNet
<i>Komplementzustandsnetz</i>	text alias	sekundär dokumentarisch	ComplementStateNet	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf ComplementStateNet
<i>Komplement-GNS-Datum</i>	text alias	primär	ComplementGNS	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf ComplementGNS

**Tabelle 73: Register I: Objekt- und Aliasregister – Teil 4**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>CompGNS</i>	export alias	sekundär exportiert	ComplementGNS	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevCompGNS:=ComplementGNS
$(\pi_\omega, \mathcal{H}_\omega, \Omega_\omega)$	composite tuple form	sekundär dokumentarisch	ComplementGNS	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	GNS-Kontext	nur für ComplementGNS; kein zweites Objekt
<i>zyklisch-separierende Lage</i>	text alias	primär	SeparatingProperty	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	Definitionskatalog / AQFT	identischer Verweis auf SeparatingProperty
<i>CycSep</i>	export alias	sekundär exportiert	SeparatingProperty	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevCycSep:=SeparatingProperty
<i>Tomita-Takesaki-Datum</i>	text alias	primär	TomitaTakesakiData	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	Definitionskatalog / AQFT	identischer Verweis auf TomitaTakesakiData
<i>TTData</i>	export alias	sekundär exportiert	TomitaTakesakiData	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevTTData:=TomitaTakesakiData
$(S, J, \Delta)$	composite tuple form	sekundär dokumentarisch	TomitaTakesakiData	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	modularer Kontext	nur für TomitaTakesakiData; kein zweites Objekt
<i>Komplement-KMS-Zustand</i>	text alias	primär	ComplementKMS	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	Definitionskatalog / AQFT	identischer Verweis auf ComplementKMS
<i>CompKMS</i>	export alias	sekundär exportiert	ComplementKMS	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevCompKMS:=ComplementKMS
$\omega_\beta$	symbol	sekundär dokumentarisch	ComplementKMS	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	KMS-Kontext	nur im KMS-Kontext für ComplementKMS
<i>Komplement-Spektralbedingung</i>	text alias	primär	ComplementSpectrumCondition	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	Definitionskatalog / AQFT	identischer Verweis auf ComplementSpectrumCondition
<i>CompSpectrum</i>	export alias	sekundär exportiert	ComplementSpectrumCondition	CNNA.AQFT	kontextgebunden; B–D bei Export opens copedCNNA.AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevCompSpectrum:=ComplementSpectrumCondition
<i>Modularkonjugation</i>	text alias	primär	ModularConjugation	CNNA.AQFT	keine Export-aktivierung; Registerform	register only	Definitionskatalog / primäre Form	identischer Verweis auf ModularConjugation
<i>ModConj</i>	export alias	sekundär exportiert	ModularConjugation	CNNA.AQFT	kontextgebunden; AQFT bei Export opens copedCNNA.AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevModConj:=ModularConjugation

**Tabelle 74: Register I: Objekt- und Aliasregister – Teil 5**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>BW-Seed</i>	text alias	primär	BisognanoWichmannSeed	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf BisognanoWichmannSeed
<i>BWSeed</i>	export alias	sekundär exportiert	BisognanoWichmannSeed	CNNA.AQFT	kontextgebunden; Registerform bei Export opens	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevBWSeed:=BisognanoWichmannSeed
<i>Nuclearity-Seed</i>	text alias	primär	NuclearitySeed	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf NuclearitySeed
<i>NucSeed</i>	export alias	sekundär exportiert	NuclearitySeed	CNNA.AQFT	kontextgebunden; Registerform bei Export opens	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevNucSeed:=NuclearitySeed
<i>DHR-Sektoren-Seed</i>	text alias	primär	DHRsectorsSeed	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf DHRsectorsSeed
<i>DHRSeed</i>	export alias	sekundär exportiert	DHRsectorsSeed	CNNA.AQFT	kontextgebunden; Registerform bei Export opens	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevDHRSeed:=DHRsectorsSeed
<i>Feldalgebra-Rekonstruktions-Seed</i>	text alias	primär	FieldAlgebraReconstructionSeed	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf FieldAlgebraReconstructionSeed
<i>FieldRecSeed</i>	export alias	sekundär exportiert	FieldAlgebraReconstructionSeed	CNNA.AQFT	kontextgebunden; Registerform bei Export opens	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevFieldRecSeed:=FieldAlgebraReconstructionSeed
<i>Ladungs-Statistik-Kohärenz</i>	text alias	primär	ChargeStatisticsCoherence	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT/Gates	register only	Definitionskatalog / AQFT	identischer Verweis auf ChargeStatisticsCoherence
<i>ChargeStatCoh</i>	export alias	sekundär exportiert	ChargeStatisticsCoherence	CNNA.AQFT	kontextgebunden; Registerform bei Export opens	AQFT/Gates	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevChargeStatCoh:=ChargeStatisticsCoherence
<i>Ladungsrekonstruktionsatz</i>	text alias	primär	ChargeReconstructionTheorem	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf ChargeReconstructionTheorem
<i>ChargeRec</i>	export alias	sekundär exportiert	ChargeReconstructionTheorem	CNNA.AQFT	kontextgebunden; Registerform bei Export opens	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevChargeRec:=ChargeReconstructionTheorem
<i>derived spacetime</i>	text alias	primär	DerivedSpacetime	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf DerivedSpacetime
<i>DerSpacetime</i>	export alias	sekundär exportiert	DerivedSpacetime	CNNA.PillarD	kontextgebunden; Registerform bei Export opens	D	exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevDerSpacetime:=DerivedSpacetime

**Tabelle 75: Register I: Objekt- und Aliasregister – Teil 6**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>Interface-Kausalität</i>	text alias	primär	InterfaceCausality	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf InterfaceCausality
<i>IfcCaus</i>	export alias	sekundär exportiert	InterfaceCausality	CNNA.PillarD	kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevIfcCaus:=InterfaceCausality
<i>lokale Kovarianz</i>	text alias	primär	LocalCovariance	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf LocalCovariance
<i>LocCov</i>	export alias	sekundär exportiert	LocalCovariance	CNNA.PillarD	kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevLocCov:=LocalCovariance
<i>Hadamard-State-Seed</i>	text alias	primär	HadamardStateSeed	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf HadamardStateSeed
<i>HadSeed</i>	export alias	sekundär exportiert	HadamardStateSeed	CNNA.PillarD	kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevHadSeed:=HadamardStateSeed
<i>mikrolokales Spektralgate</i>	text alias	primär	MicrolocalSpectrumGate	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf MicrolocalSpectrumGate
<i>MicroSpec</i>	export alias	sekundär exportiert	MicrolocalSpectrumGate	CNNA.PillarD	kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevMicroSpec:=MicrolocalSpectrumGate
<i>lokale Wick-Polynomseed</i>	text alias	primär	LocalWickPolynomialsSeed	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf LocalWickPolynomialsSeed
<i>WickSeed</i>	export alias	sekundär exportiert	LocalWickPolynomialsSeed	CNNA.PillarD	kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevWickSeed:=LocalWickPolynomialsSeed
<i>Einstein-Limes-Seed</i>	text alias	primär	EinsteinLimitSeed	CNNA.PillarD	keine Export-aktivierung; Registerform	Pillar D	register only	Definitionskatalog / Pillar D	identischer Verweis auf EinsteinLimitSeed
<i>EinsteinSeed</i>	export alias	sekundär exportiert	EinsteinLimitSeed	CNNA.PillarD	kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevEinsteinSeed:=EinsteinLimitSeed
<i>Zeitumkehr auf GNS</i>	text alias	primär	TimeReversalOnGNS	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf TimeReversalOnGNS
<i>TRonGNS</i>	export alias	sekundär exportiert	TimeReversalOnGNS	CNNA.AQFT	kontextgebunden; AQFT bei Export opens		exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevTRonGNS:=TimeReversalOnGNS

**Tabelle 76: Register I: Objekt- und Aliasregister – Teil 7**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>Spinrepräsentations-Seed</i>	text alias	primär	SpinRepresentationSeed	CNNA.AQFT	keine Export-aktivierung;	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf SpinRepresentationSeed
<i>SpinSeed</i>	export alias	sekundär exportiert	SpinRepresentationSeed	CNNA.AQFT	Registerform kontextgebunden;	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevSpinSeed:=SpinRepresentationSeed
<i>Statistik-Gate</i>	text alias	primär	StatisticsGate	CNNA.AQFT	keine Export-aktivierung;	AQFT/Gates	register only	Definitionskatalog / AQFT	identischer Verweis auf StatisticsGate
<i>StatGate</i>	export alias	sekundär exportiert	StatisticsGate	CNNA.AQFT	Registerform kontextgebunden;	AQFT/Gates	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevStatGate:=StatisticsGate
<i>Spin-Statistik-Theorem</i>	text alias	primär	SpinStatisticsTheorem	CNNA.AQFT	keine Export-aktivierung;	AQFT	register only	Definitionskatalog / AQFT	identischer Verweis auf SpinStatisticsTheorem
<i>SpinStatThm</i>	export alias	sekundär exportiert	SpinStatisticsTheorem	CNNA.AQFT	Registerform kontextgebunden;	AQFT	exported	Notations-Appendix / CNNA/AQFT/Notation.lean	scopedabbrevSpinStatThm:=SpinStatisticsTheorem
<i>PT-Bridge-Seed</i>	text alias	primär	PTBridgeSeed	CNNA.PillarE	keine Export-aktivierung;	D–E	register only	Definitionskatalog / Pillar E	identischer Verweis auf PTBridgeSeed
<i>PTSeed</i>	export alias	sekundär exportiert	PTBridgeSeed	CNNA.PillarE	Registerform kontextgebunden;	D–E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevPTSeed:=PTBridgeSeed
<i>Ladungskonjugations-Seed</i>	text alias	primär	ChargeConjugationSeed	CNNA.PillarE	keine Export-aktivierung;	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf ChargeConjugationSeed
<i>CSeed</i>	export alias	sekundär exportiert	ChargeConjugationSeed	CNNA.PillarE	Registerform kontextgebunden;	E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevCSeed:=ChargeConjugationSeed
<i>CPT-Gate</i>	text alias	primär	CPTGate	CNNA.PillarE	keine Export-aktivierung;	B–D–E	register only	Definitionskatalog / Gates	identischer Verweis auf CPTGate
<i>CPTCheck</i>	export alias	sekundär exportiert	CPTGate	CNNA.PillarE	Registerform kontextgebunden;	B–D–E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevCPTCheck:=CPTGate
<i>CPT-Theorem</i>	text alias	primär	CPTTheorem	CNNA.AQFT	keine Export-aktivierung;	B–D–E	register only	Definitionskatalog / AQFT	identischer Verweis auf CPTTheorem
<i>CPTThm</i>	export alias	sekundär exportiert	CPTTheorem	CNNA.AQFT	Registerform kontextgebunden;	B–D–E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevCPTThm:=CPTTheorem

**Tabelle 77: Register I: Objekt- und Aliasregister – Teil 8**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>Kramers-Gate</i>	text alias	primär	KramersGate	CNNA.PillarE	keine Export-aktivierung; Registerform	AQFT/E	register only	Definitionskatalog / Gates	identischer Verweis auf KramersGate
<i>KramersCheck</i>	export alias	sekundär exportiert	KramersGate	CNNA.PillarE	kontextgebunden; bei Export opens	AQFT/E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevKramersCheck:=KramersGate
<i>Kramers-Theorem</i>	text alias	primär	KramersTheorem	CNNA.AQFT	keine Export-aktivierung; Registerform	AQFT/E	register only	Definitionskatalog / AQFT	identischer Verweis auf KramersTheorem
<i>KramersThm</i>	export alias	sekundär exportiert	KramersTheorem	CNNA.AQFT	kontextgebunden; bei Export opens	AQFT/E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevKramersThm:=KramersTheorem
<i>Gauge-Sektor-Seed</i>	text alias	primär	GaugeSectorSeed	CNNA.PillarE	keine Export-aktivierung; Registerform	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf GaugeSectorSeed
<i>GaugeSeed</i>	export alias	sekundär exportiert	GaugeSectorSeed	CNNA.PillarE	kontextgebunden; bei Export opens	E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevGaugeSeed:=GaugeSectorSeed
<i>Anomaly-Inflow-Seed</i>	text alias	primär	AnomalyInflowSeed	CNNA.PillarE	keine Export-aktivierung; Registerform	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf AnomalyInflowSeed
<i>InflowSeed</i>	export alias	sekundär exportiert	AnomalyInflowSeed	CNNA.PillarE	kontextgebunden; bei Export opens	E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevInflowSeed:=AnomalyInflowSeed
<i>Anomalie-Kompensations-Gate</i>	text alias	primär	AnomalyCancellationGate	CNNA.PillarE	keine Export-aktivierung; Registerform	E/Gates	register only	Definitionskatalog / Gates	identischer Verweis auf AnomalyCancellationGate
<i>AnomCancel</i>	export alias	sekundär exportiert	AnomalyCancellationGate	CNNA.PillarE	kontextgebunden; bei Export opens	E/Gates	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevAnomCancel:=AnomalyCancellationGate
<i>Ladungsquantisierungs-Gate</i>	text alias	primär	ChargeQuantizationGate	CNNA.PillarE	keine Export-aktivierung; Registerform	E/Gates	register only	Definitionskatalog / Gates	identischer Verweis auf ChargeQuantizationGate
<i>ChargeQuant</i>	export alias	sekundär exportiert	ChargeQuantizationGate	CNNA.PillarE	kontextgebunden; bei Export opens	E/Gates	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevChargeQuant:=ChargeQuantizationGate
<i>Higgs-Sektor-Seed</i>	text alias	primär	HiggsSectorSeed	CNNA.PillarE	keine Export-aktivierung; Registerform	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf HiggsSectorSeed
<i>HiggsSeed</i>	export alias	sekundär exportiert	HiggsSectorSeed	CNNA.PillarE	kontextgebunden; bei Export opens	E	exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevHiggsSeed:=HiggsSectorSeed

**Tabelle 78: Register I: Objekt- und Aliasregister – Teil 9**

Oberflächenform	Formtyp	Formrang	kanonischer Lean-Name	Namespace	Aktivierungsregel	Scope	Exportstatus	bindende Stelle	Expansion / Verweisregel
<i>EWSB-Seed</i>	text alias	primär	EWSBSeed	CNNA.PillarE	keine Export-aktivierung;	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf EWSBSeed
<i>EWSBSeed</i>	export alias	sekundär exportiert	EWSBSeed	CNNA.PillarE	Registerform kontextgebunden; E bei Export opens		exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevEWSBSeed:=EWSBSeed
<i>Yukawa-Flavor-Seed</i>	text alias	primär	YukawaFlavorSeed	CNNA.PillarE	keine Export-aktivierung;	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf YukawaFlavorSeed
<i>YukawaSeed</i>	export alias	sekundär exportiert	YukawaFlavorSeed	CNNA.PillarE	Registerform kontextgebunden; E bei Export opens		exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevYukawaSeed:=YukawaFlavorSeed
<i>Generationen-Seed</i>	text alias	primär	GenerationSeed	CNNA.PillarE	keine Export-aktivierung;	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf GenerationSeed
<i>GenSeed</i>	export alias	sekundär exportiert	GenerationSeed	CNNA.PillarE	Registerform kontextgebunden; E bei Export opens		exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevGenSeed:=GenerationSeed
<i>SM-Matching-Window</i>	text alias	primär	SMMatchingWindow	CNNA.PillarE	keine Export-aktivierung;	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf SMMatchingWindow
<i>SMMatch</i>	export alias	sekundär exportiert	SMMatchingWindow	CNNA.PillarE	Registerform kontextgebunden; E bei Export opens		exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevSMMatch:=SMMatchingWindow
<i>SMEFT-Window</i>	text alias	primär	SMEFTWindow	CNNA.PillarE	keine Export-aktivierung;	E	register only	Definitionskatalog / Pillar E	identischer Verweis auf SMEFTWindow
<i>SMEFTWin</i>	export alias	sekundär exportiert	SMEFTWindow	CNNA.PillarE	Registerform kontextgebunden; E bei Export opens		exported	Notations-Appendix / CNNA/PillarE/Notation.lean	scopedabbrevSMEFTWin:=SMEFTWindow
<i>Dirac-Operator-Seed</i>	text alias	primär	DiracOperatorSeed	CNNA.PillarD	keine Export-aktivierung;	D	register only	Definitionskatalog / Pillar D	identischer Verweis auf DiracOperatorSeed
<i>DiracSeed</i>	export alias	sekundär exportiert	DiracOperatorSeed	CNNA.PillarD	Registerform kontextgebunden; D bei Export opens		exported	Notations-Appendix / CNNA/PillarD/Notation.lean	scopedabbrevDiracSeed:=DiracOperatorSeed

## F.2 Register II: Symbol- und Parameterregister

**Dokumentebene:** Meta-Audit-Ebene

Dieses Register enthält nur Symbole und Parameter. Besonders wichtig ist die explizite Aufspaltung der bisher überladenen  $\Omega$ -Verwendung in einen zyklischen Vektor  $\Omega_\omega$  und einen Carrier-/Indexkontext  $\Omega_{\text{idx}}$ .

**Tabelle 79: Register II: Symbol- und Parameterregister – Teil 1**

Symbol	Symbolklasse	kanonische Bedeutung	zulässiger Kontext	verbotener Kontext	Disambiguierungsregel	Ersatz- / Migrationsre- bindende Stelle	Status
$b$	parameter	Branching-/Mikrostrukturparameter	Closure-/A-Kontext	niemals ohne expliziten Closure-Verweis	nur für Branching-/Mikrostrukturparameter	später durch derived Selektoren oder Closure ersetzt	Parameter-Closure-Strang später derived oder eliminiert
$L_{\max}$	legacy parameter	alter freier IR-/Horizont-Parameter	nur in Archiv- und Migrationskontexten	im aktiven Kern als Primärname verboten	nur als Legacy-Cutoffsymbol	wird langfristig durch $\text{HorizonLevel} / L_H$ ersetzt	Migration nach $\text{HorizonLevel}$ Legacyname; aktiv nicht normativ
$L_H$	derived scale	$\text{HorizonLevel}$	C–D	nicht für $L_{\max}$ verwenden	nie als freies $L_{\max}$ lesen	ersetzt aktiven Gebrauch von $L_{\max}$	Pillar D / $\text{HorizonLevel}$ aktive derived Skala
$\beta$	KMS parameter	inverse Temperatur bzw. KMS-Parameter	KMS-/Closure-Kontext	nicht unmarkiert für beliebige andere Parameter	nur als KMS-/Closure-Parameter	nicht unmarkiert auf andere Parameter übertragen	B-Kern und Closure-Diskussion standardnah, aber closure-pflichtig
$\omega_{\text{comp}}$	state symbol	$\text{ComplementStateNet}$	AQFT-Zustandskontext	nicht für $\text{ComplementKMS}$	nur für $\text{ComplementStateNet}$	kein Ersatz für $\omega_\beta$	$\text{ComplementStateNet}$ symbolisch erlaubt
$\omega_\beta$	KMS state symbol	$\text{ComplementKMS}$	nur KMS-Kontext	nicht für das allgemeine Zustandsnetz	nur im KMS-Kontext	nie als Symbol für das allgemeine Zustandsnetz	$\text{ComplementKMS}$ symbolisch erlaubt
$C$	sector-family symbol	$\text{ComplementSectorFamily}$	nur Komplementfamilien-Kontext	nicht für beliebige Kategorien oder Konstanten	nur für die Komplementsektorfamilie des CNNA-Kerns	nicht als generisches C-Symbol ausserhalb des CNNA-Kontexts verwenden	$\text{ComplementSectorFamily}$ kontextgebunden
$\Sigma_{\text{BDI}}$	sector-split symbol	$\text{SectorSplit}$	nur Bright/Dark/Interface-Split-Kontext	nicht für beliebige Summen- oder Schnittoperationen	nur für den sektor-first-Split des CNNA-Kerns	keine Wiederverwendung als allgemeines Sigma-Schema	$\text{SectorSplit}$ kontextgebunden
$\Lambda_{\text{gen}}$	DtN core symbol	$\text{GeneralizedDtN}$	nur DtN-/Schur-/Eliminationskontext	nicht für beliebige Spektral- oder Koppelungsoperatoren	nur für den generalisierten DtN-Kern	nicht mit anderen Lambda-Parametern oder Spektren mischen	$\text{GeneralizedDtN}$ kontextgebunden
$\mathcal{A}_{\text{comp}}$	net symbol	$\text{ComplementLocalNet}$	nur AQFT-Netzkontext	nicht für beliebige Algebrfamilien ausserhalb des Komplementnetzes	nur für das Komplementnetz	kein Ersatz für Interface- oder Bright-Netze	$\text{ComplementLocalNet}$ symbolisch erlaubt
$\mathcal{A}_{\text{int}}$	net symbol	$\text{InterfaceLocalNet}$	nur Interface-/AQFT-Netzkontext	nicht für das Komplementnetz oder allgemeine Algebrfamilien	nur für das Interface-netz	kein Ersatz für $\mathcal{A}_{\text{comp}}$	$\text{InterfaceLocalNet}$ symbolisch erlaubt
$\pi_\omega$	representation symbol	GNS-Darstellung aus $\text{ComplementGNS}$	nur GNS-/Darstellungskontext	nicht für beliebige projektive Abbildungen	nur für die im GNS-Datum gebundene Darstellung	nicht als freies pi-Symbol ausserhalb des GNS-Kontexts lesen	$\text{ComplementGNS}$ kontextgebunden
$\mathcal{H}_\omega$	Hilbert-space symbol	Hilbertraum des GNS-Datums	nur GNS-/Darstellungskontext	nicht für beliebige Hilberträume ohne GNS-Bindung	nur für den im GNS-Datum gebundenen Hilbertraum	nicht als allgemeines H-Symbol ohne Datum verwenden	$\text{ComplementGNS}$ kontextgebunden
$\Omega_\omega$	cyclic vector	zyklischer Vektor des GNS-Datums	nur GNS-Kontext	nicht für Carrierindexmengen	nur für zyklischen Vektor im GNS-/KMS-Kontext	nie für Carrierindex verwenden	$\text{ComplementGNS}$ kontextgebunden
$\Omega_{\text{id}_x}$	carrier/index symbol	Index- oder Trägermenge	nur Carrier-/Netzkontext	nicht für zyklischen Vektor	nur für Index-/Carrierkontext	ersetzt alte überladene $\Omega$ -Verwendung für Carrier	Carrier-/Netzdefinitionen Disambiguierungersatz für überladenes $\Omega$
$S$	Tomita operator symbol	Tomita-Operator	nur Tomita–Takesaki-Kontext	nicht für beliebige Wirkungsfunktionen, Mengen oder S-Matrizen	nur für den Tomita-Operator des modularen Datums	in Streu- oder Wirkungskontexten nicht ohne Index/Klarstellung verwenden	$\text{TomitaTakesakiData}$ symbolisch erlaubt
$J$	modular symbol	modulare Konjugation	nur Tomita–Takesaki-Kontext	nicht als beliebiges lineares Objekt	nur für modulare Konjugation	nicht als beliebiges lineares Objekt verwenden	$\text{TomitaTakesakiData}$ ModularConjugation symbolisch erlaubt
$\Delta$	modular symbol	modularer Operator	nur Tomita–Takesaki-Kontext	nicht als Differenzenoperator o. a.	nur für modularen Operator	nicht als Differenzenoperator o. a. lesen	$\text{TomitaTakesakiData}$ symbolisch erlaubt
$\Theta$	time-reversal symbol	implementierte Zeitumkehr	nur Zeitumkehr-/Kramers-Kontext	nicht für beliebige Involutionen	nur für implementierte Zeitumkehr	nicht für beliebige Involutionen verwenden	$\text{TimeReversalOnGNS}$ KramersGate kontextgebunden
$(-1)^F$	fermion parity symbol	Fermionparität	nur gradiert Statistik-/Kramers-Kontext	nicht für allgemeine Signaturen	nur für Fermionparität	nicht für allgemeine Signaturen oder Gradiertungen ausserhalb des Statistikpfads	$\text{GradedStarAlgebra}$ KramersGate kontextgebunden

### **F.3 Register III: Reifegrad- und Architekturstatusregister**

**Dokumentebene:** Meta-Audit-Ebene

Reifegradmarker sind keine Fachobjekte. Sie dürfen deshalb nicht in Objekt- oder Symbolregister eingemischt werden.

**Tabelle 80: Register III: Reifegrad- und Architekturstatusregister – Teil 1**

Marker	Bedeutung	zulässig für	unzulässig für	bindende Regel	Migrationspflicht	Status
<b>Seed</b>	frühe, noch nicht abgeschlossene Vorzone	def, structure, theorem route, gate route	Endtheoreme, vollständige Closure-Objekte, reine Meta-Begriffe	Regelwerk / Reifegradkonvention	muss später de-seeded, geschlossen oder explizit als dauerhaft früh markiert werden	globaler Statusmarker
<b>Scaffold</b>	strukturell getragener, aber noch nicht voll geschlossener Träger	structure, interface carrier, transition object	blosse Prosa-Begriffe, reine Reviewerrollen	Regelwerk / Reifegradkonvention	muss entweder in aktiven Kern übergehen oder als Zwischenstufe bestehen bleiben	globaler Statusmarker
<b>Window</b>	spätes Spezial- oder Matching-Fenster ausserhalb des Kerns	phenomenology windows, IR windows, matching windows	active core, grundlegende Stammobjekte, zwingende Closure-Gates	Regelwerk / Reifegradkonvention	darf nicht stillschweigend in Kernontologie umetikettiert werden	globaler Statusmarker

#### **F.4 Register IV: Meta-Methodikregister**

**Dokumentebene:** Meta-Audit-Ebene

Meta-Begriffe sind dokumentations- und auditrelevant, aber nicht Bestandteil der fachlichen Notations-Emission. Deshalb werden sie getrennt registriert.

Tabelle 81: Register IV: Meta-Methodikregister – Teil 1

Meta-Begriff	Bedeutung	Dokumenttyp	Lean-Objekt?	kanonischer Lean-Name / Ableitungs- verbot	Ableitungs- Ableitungsregel	bindende Stelle	Status
TruthProtocol	mehrstufiges Wahrheits- und Falsifikationsprotokoll	Roadmap, Spec, Audit-Anhang	nein	kein Lean-Objekt; Ableitungsverbot	keine Notationsableitung in <code>Notation.lean</code>	Methodik-Abschnitt	meta only
ExpertReview	unverzichtbarer externer Fachreview jenseits der Maschinenprüfung	Roadmap, Audit, Review-Prozess	nein	kein Lean-Objekt; Ableitungsverbot	keine Notationsableitung in <code>Notation.lean</code>	Methodik-Abschnitt	meta only

**Integritätsregeln der Lesbarkeitsschicht.** Es gelten global:

- (1) Kein Appendix-Eintrag ohne Definitionskatalog-Anker.
- (2) Kein Registereintrag ohne kanonischen Zielanker oder ausdrückliches Ableitungsverbot.
- (3) Keine exportierte Form ohne Namespace- und Aktivierungsregel.
- (4) Keine Symbolverwendung ohne Disambiguierungsregel.
- (5) Keine sekundäre Exportform ohne expliziten Formrang.
- (6) Keine Meta-Begriffe in objektbezogenen Notationstabellen.
- (7) Keine Rückdelegation wesentlicher Tabellensemantik an proseartige Erläuterungen.

## G Instanz- und Kontextblatt für tragende Pfade

Dokumentebene: Meta-Audit-Ebene

Tabelle 82: Instanz- und Kontextblatt für tragende Pfade – Teil 1

Pfad / Objekt	globale Voraussetzungen	lokale Voraussetzungen	Instanz-/Kontexttyp	Dokumentrolle
GeneralizedDtN	ToC-Vertrag, SectorSplit	Glueing-/Restriktionsdaten	strukturiertes Exportkontext	A→B-Handoff
ComplementLocalNet	Regionen, SectorSplit, DtN-Daten	Netzlokalisierung	AQFT-Kontext	Start des B-Kerns
ComplementGNS	ComplementStateNet	GNS-taugliche Zustandslage	Darstellungs-/Hilbertkontext	Beginn modularer Kette
TomitaTakesakiData	GNS-Lage	SeparatingProperty	operatoralgebraischer Kontext	B-Kern-Mitte
ComplementSpectrumCondition	KMS/positive Energie	Vorwärtskegelkontrolle	Spektral-/Generator-Kontext	Vorbedingung für BW/Scattering
ParameterClosure	Kanal-/Entropiedaten	Selektoren, Fixpunktstruktur	pfeilerübergreifender Closure-Kontext	C→D-Schwelle
HadamardStateSeed	LocalCovariance	gekrümmter Hintergrund / Mikrolokalität	LCQFT-Kontext	renormierte D-Sprache
FieldAlgebraReconstructionSeed	DHR-Sektoren	Charge-/Statistik-Kohärenz	rekonstruktiver Sektorkontext	B/E-Vorzone

## H Prüfprotokoll der Roadmap

**Dokumentebene:** Meta-Audit-Ebene

- Geprüfte Codebasis: archivgebunden an `REAL-OQS_v0.0605_full_no_comments.zip`.
- Geprüfter Dokumenttyp: Projekt-/Pfad-Roadmap mit Handoff-, Ketten- und Audit-Anhängen; keine Einzeltheorem-Voll-Dokumentation.
- Geprüfte Trennungen: CNNA-Kern vs. Bright-Recovery; Seed/Scaffold/Window vs. de-seedeter Kern; Dokumentation vs. physikalische Interpretation.
- Sichtbare Auditwerkzeuge: Strukturkarte, partielle Ordnung, Gates, Handoffs, Kettenblätter, Register, Kontextblatt, Meta-Audit-Marker.
- Umgang mit `sorry/admit`/freien Axiomen: im kritischen Pfad nicht zulässig; die Roadmap setzt formale Verifikation statt rhetorischer Ersatzbegründung voraus.
- Umgang mit privaten Hilfssätzen und `omit`-Kontexten: müssen später in Datei-/Theorem-Dokumentationen explizit nachgetragen werden; hier nur als Auditpflicht markiert, nicht unsichtbar vorausgesetzt.
- Delta-Fokus dieser Fassung: Ergänzung regelwerksnaher Metadaten, Ebenenmarken, Definitionskatalog, Handoff-Spezifikationen, Kettenblätter, Register und Audit-Anhänge.

## I Namens- und Pfadverzeichnis

### Dokumentenebene: Meta-Audit-Ebene

Tabelle 84: Namens- und Pfadverzeichnis – Teil 1

Name / Pfadbereich	Rolle in der Roadmap
CNNA/Core/*	ontischer und regionslogischer Kern, einschliesslich ToC-Vertrag, Sector-Split und InfiniteCarrier
CNNA/OQS/*	GeneralizedDtN-, MultiSchur-, Kanal- und Rückwirkungsstrang
CNNA/AQFT/*	lokale Netze, Zustände, GNS, modulare Schicht, DHR/DR-Vorzone
CNNA/PillarD/*	emergente Geometrie, LocalCovariance, Relative Cauchy Evolution, Gravity-Routen
CNNA/PillarE/*	späte CPT-/Statistik-/Gauge-/Matter-/SM-Schicht
CNNA/AQFT/BrightRecovery/*	expliziter Recovery-/Regressionstrang des alten Boundary-Pfads
REALOQS/PillarA/*	wichtigste Quellbasis für ontische und OQS-nahe Portierungen
REALOQS/PillarB/*	wichtigste Quellbasis für AQFT-, Modularitäts- und Bright-Recovery-Altpfade
REALOQS/PillarC/*	wichtigste Quellbasis für finite Kanäle, Kraus-/RectKraus-Strukturen und Rückwirkung

## J Meta-Audit-Marker

### Dokumentebene: Meta-Audit-Ebene

Tabelle 86: Meta-Audit-Marker – Teil 1

Marker	Bedeutung
M0	Dieses Dokument ist archivgebunden an REAL-OQS_v0.0605_full_no_comments.zip; nicht commitgebunden.
M1	Kein boundary-first Dateiname darf auf dem kritischen CNNA-Pfad Architekturzentrum sein.
M2	$L_{\max}$ ist auf dem aktiven Pfad kein ontischer Grundparameter, sondern höchstens Scaffold-Name vor HorizonLevel.
M3	Der alte BoundaryMatrix-Pfad ist Bright-Spezialfall und Regressionsorakel, nicht Kernsemantik.
M4	D und E dürfen nur aus expliziten Source-Maps und abgeleiteten Entropie-/Spektral-/Backreactiondaten de-seeden.
M5	Prototypisches Vorseilen ist nur als Seed/Scaffold zulässig; frühe Dummy-Schnittstellen prüfen Typtragefähigkeit, nicht Kernclosure.
M6	Rückfluss später Einsichten in frühere Pfeiler ist erlaubt, aber nur über kontrolliertes Re-Opening, Regressionsprüfung und erneutes Gate-Closing.
M7	AI ist auf dem kritischen Pfad nur Heuristik- und Übersetzungsschicht; technische Wahrheitsquelle bleiben formale Sprache, Kernel-Check und die explizite Disziplin gegen sorry/admit/freie Axiome.
M8	Formale Korrektheit des Codes garantiert noch keine korrekte Prosa-Semantik oder physikalische Relevanz; Kommentar-, Dokument- und Interpretationsprüfung sind eigener Pflichtenritt.
M9	Spezialfenster (Minkowski, Conformal, FLRW, SM, SMEFT, Hadron, ChiralEFT) bleiben standardmäßig Windows.
M10	Asymptotic-Safety-Sprache ist erst nach Closure diagnostisch legitim; sie ersetzt die Closure nicht.
M11	String-mathematische Ideen werden als Organisations- und Glueing-/Defekt-/Inflow-Werkzeuge genutzt, nicht als neues fundamentales Postulat.

## K Tabellarischer Definitionskatalog der Kernobjekte

**Dokumentebene:** Projekt-Ebene und Pfad-Ebene

Dieser Anhang enthält den aus dem Haupttext ausgelagerten vollständigen tabellarischen Definitionskatalog. Er bleibt die kanonische Objektquelle der Roadmap; die normativen Regeln hierzu stehen weiterhin in Abschnitt 4.

**Tabelle K.1-A1: Definitionskatalog der Kernobjekte – Objektbasis, Teil 1**

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
TreeOfCliques	structure	<i>Tree of Cliques</i>	<i>ToC</i>	text alias	<i>Tree of Cliques</i> und <i>ToC</i> verweisen identisch auf <i>TreeOfCliques</i> ; keine konkurrierende Zweitbedeutung	CNNA/Core	CNNA/Notation.lean	CNNA	openscopedCNNA	global
BranchPatch	structure	<i>Bright-Patch</i>	<i>Bright-Patch</i>	text alias	<i>Bright-Patch</i> expandiert auf <i>BranchPatch</i>	CNNA/Core bzw. Pillar A	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A
ComplementSectorFamily	structure	<i>C</i>	<i>C</i>	symbolic notation + text alias	<i>C</i> und <i>Komplementsektorfamilie</i> verweisen auf <i>ComplementSectorFamily</i>	CNNA/Core bzw. Pillar A/Complement	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A–B
SectorSplit	structure	$\Sigma_{BDI}$	$\Sigma_{BDI}$	symbolic notation + text alias	$\Sigma_{BDI}$ und <i>Bright/Dark/Interface-Split</i> verweisen auf <i>SectorSplit</i>	CNNA/Core/Sector-Split	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A–B
GeneralizedDtN	def	$\Lambda_{gen}$	$\Lambda_{gen}$	symbolic notation + text alias	$\Lambda_{gen}$ und <i>generalisierter DtN-Kern</i> verweisen auf <i>GeneralizedDtN</i>	CNNA/OQS/GeneralizedDtN	CNNA/PillarA/Notation.lean	CNNA.PillarA	openscopedCNNA.PillarA	Pillar A–B
SectorChannels	structure	<i>Sektorkanäle</i>	<i>SectCh</i>	text alias	<i>Sektorkanäle</i> verweisen auf <i>SectorChannels</i>	CNNA/OQS	CNNA/OQS/Notation.lean	CNNA.OQS	openscopedCNNA.OQS	Pillar C
SectorSysEnv	structure	<i>Sektor-SysEnv-Split</i>	<i>SysEnvSplit</i>	text alias	<i>Sektor-SysEnv-Split</i> verweist auf <i>SectorSysEnv</i>	CNNA/OQS	CNNA/OQS/Notation.lean	CNNA.OQS	openscopedCNNA.OQS	Pillar C
RelativeEntropyFlow	def	<i>relativer Entropiefluss</i>	<i>RelEntFlow</i>	text alias	<i>relativer Entropiefluss</i> verweist auf <i>RelativeEntropyFlow</i>	CNNA/OQS bzw. Pillar C	CNNA/OQS/Notation.lean	CNNA.OQS	kontextgebunden; bei Export <i>openscopedCNNA.OQS</i>	Pillar C–D
GlobalStateCone	structure	<i>GlobalStateCone</i>	—	selfspeaking	die lesbare Form ist identisch mit dem Lean-Namen; keine Lichtkegel-Bedeutung	pfeilerübergreifend C–E	kein eigener Aliasexport	CNNA	keine Exportaktivie-global	rung
Backreaction	def	<i>Backreaction</i>	—	selfspeaking	identisch mit dem Lean-Namen	pfeilerübergreifend C–D	kein eigener Aliasexport	CNNA	keine Exportaktivie-global	rung
ComplementClosure	structure	<i>Komplement-schliessung</i>	<i>CompClosure</i>	text alias	<i>Komplement-schliessung</i> verweist auf <i>ComplementClosure</i>	AQFT-/Closure-Pfad	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscopedCNNA.AQFT</i>	AQFT/C

Tabelle K.1-B1: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 1

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
TreeOfCliques	Exportform reserviert; keine konkurrierende Kurzform		BranchPatch, ComplementSectorFamily, SectorSplit	active core	ontischer Startträger der Stammtheorie
BranchPatch	Exportform reserviert; keine konkurrierende Kurzform	TreeOfCliques	ComplementSectorFamily, SectorSplit, GeneralizedDtN	active core	erster expliziter sichtbarer Patch relativ zum ToC-Anker
ComplementSectorFamily	nur im angegebenen Namespace; keine Zweitebelegung der Symbolform	TreeOfCliques, BranchPatch	SectorSplit, ComplementLocalNet, ComplementStateNet	active core	organisierte Familie der dunklen, trunk- und seitenastbezogenen Komplemente
SectorSplit	nur im angegebenen Namespace; keine Zweitebelegung der Symbolform	TreeOfCliques, ComplementSectorFamily	GeneralizedDtN, SectorChannels, ComplementLocalNet	active core	synchronisierte Dreiteilung des sektor-first Pfads
GeneralizedDtN	nur im angegebenen Namespace; keine Zweitebelegung der Symbolform	SectorSplit, ComplementSectorFamily	SectorChannels, ComplementLocalNet, InterfaceLocalNet	active core	Kopplungs- und Eliminationskern zwischen Bright-, Dark- und Interface-Sektor
SectorChannels	Exportform reserviert; keine konkurrierende Kurzform	GeneralizedDtN, SectorSplit	SectorSysEnv, RelativeEntropyFlow, Backreaction	active extension	Kanalstruktur für Bright/System/Environment- bzw. Mehrsektor-Lesarten
SectorSysEnv	Exportform reserviert; keine konkurrierende Kurzform	SectorChannels	RelativeEntropyFlow, ComplementStateNet	active extension	explizite System/Environment-Schicht über dem sektorischen Grundbild
RelativeEntropyFlow	Exportform reserviert; keine konkurrierende Kurzform	SectorChannels, SectorSysEnv	Backreaction, EffectiveLambda, BackreactionFixedPoint	active extension	koppelt Zustandsänderung, Entropie und Rückwirkung
GlobalStateCone	kein Exportalias; Lean-Name reserviert	ComplementStateNet, ComplementKMS	Backreaction, RelativeEntropyFlow	active extension	globaler Zustandskegel, nicht physikalischer Raumzeitkegel
Backreaction	kein Exportalias; Lean-Name reserviert	RelativeEntropyFlow, GlobalStateCone	EffectiveLambda, BackreactionFixedPoint, DerivedSpacetime	active extension	Rückkopplung von Zustand, Entropie, Spektrum und Skale
ComplementClosure	Exportform reserviert; keine konkurrierende Kurzform	ComplementLocalNet, ComplementStateNet	ComplementGNS, InfiniteCarrier	active extension	Schliessungsobjekt des Komplementnetzes vor weiterem Limes- oder Quasilokalitätsübergang

**Tabelle K.1-A2: Definitionskatalog der Kernobjekte – Objektbasis, Teil 2**

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
ParameterClosure	gate	<i>Parameter-Closure</i>	<i>ParamClosure</i>	text alias	<i>Parameter-Closure</i> verweist auf ParameterClosure	pfeilerübergreifend C-D	CNNA/Notation.lean	CNNA	openscopedCNNA	global
EffectiveLambda	def	$\lambda_{\text{eff}}$	$\lambda_{\text{eff}}$	symbolic notation + text alias	$\lambda_{\text{eff}}$ und <i>EffectiveLambda</i> verweisen auf EffectiveLambda	Closure-/D-Pfad	CNNA/PillarD/Notation.lean	CNNA.PillarD	openscopedCNNA.PillarD	C-D
RegularizationClosure	gate	<i>Regularisierungs-Closure</i>	<i>RegClosure</i>	text alias	<i>Regularisierungs-Closure</i> verweist auf RegularizationClosure	Closure-Pfad	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope dCNNA.PillarD	C-D
BranchingWitness	theorem witness	<i>Branching-Witness</i>	<i>BranchWit</i>	text alias	<i>Branching-Witness</i> verweist auf BranchingWitness	Pillar A/D-Closure	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscope dCNNA.PillarA	A-D
selectedBranching	def	<i>selected branching</i>	<i>SelBranch</i>	text alias	<i>selected branching</i> verweist auf selectedBranching	Closure-/Selektorpfad	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscope dCNNA.PillarA	A-D
UVSpectralSelector	selector	<i>UV-Spektralselektor</i>	<i>UVSel</i>	text alias	<i>UV-Spektralselektor</i> verweist auf UVSpectralSelector	Closure-/Selektorpfad	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscope dCNNA.PillarA	A-D
BranchingSelector	selector	<i>Branching-Selektor</i>	<i>BranchSel</i>	text alias	<i>Branching-Selektor</i> verweist auf BranchingSelector	Closure-/Selektorpfad	CNNA/PillarA/Notation.lean	CNNA.PillarA	kontextgebunden; bei Export openscope dCNNA.PillarA	A-D
BackreactionSelector	selector	<i>Backreaction-Selektor</i>	<i>BackSel</i>	text alias	<i>Backreaction-Selektor</i> verweist auf BackreactionSelector	Closure-/Selektorpfad	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope dCNNA.PillarD	C-D
InfiniteCarrier	structure	InfiniteCarrier	—	selfspeaking	identisch mit dem Namen; keine freie Synonymbildung	AQFT-/Quasilokalitäts- und D-Limespfad	kein eigener Aliasexport	CNNA	keine Exportaktivierung	B-D
HorizonLevel	def	$L_H$	$L_H$	symbolic notation + text alias	$L_H$ und <i>Horizonniveau</i> verweisen auf HorizonLevel	CNNA/PillarD/HorizonLevel	CNNA/PillarD/Notation.lean	CNNA.PillarD	openscopedCNNA.PillarD	C-D
BackreactionFixedPoint	gate	<i>Backreaction-Fixpunkt</i>	<i>BackFix</i>	text alias	<i>Backreaction-Fixpunkt</i> verweist auf BackreactionFixedPoint	CNNA/PillarD/BackreactionFixedPoint	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope dCNNA.PillarD	C-D

Tabelle K.1-B2: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 2

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
ParameterClosure	Exportform reserviert; keine konkurrie-RelativeEntropyFlow, Backreaction rende Kurzform		EffectiveLambda, HorizonLevel, Backreact ionFixedPoint	active core	derived-only-Schliessung freier Restparameter
EffectiveLambda	nur im angegebenen Namespace; keine Backreaction, ParameterClosure Zweitbelegung der Symbolform		RegularizationClosure, BackreactionFixe dPoint, DerivedSpacetime	planned active	effektive Größe für rückwirkungssta- bilisierte Skalenlesarten
RegularizationClosure	Exportform reserviert; keine konkurrie-EffectiveLambda rende Kurzform		HorizonLevel, BackreactionFixedPoint, De rivedSpacetime	planned active	Schliessung von Regularisierung und derived Skala
BranchingWitness	Exportform reserviert; keine konkurrie-BranchPatch, SectorSplit rende Kurzform		selectedBranching, UVSpectralSelector, B ranchingSelector	active extensi- on	expliziter Zeuge nichttrivialer Verzweigung
selectedBranching	Exportform reserviert; keine konkurrie-BranchingWitness rende Kurzform		UVSpectralSelector, BranchingSelector, B ackreactionSelector	active extensi- on	ausgewählte Verzweigung nach dokumentierter Selektorlogik
UVSpectralSelector	Exportform reserviert; keine konkurrie-selectedBranching rende Kurzform		BranchingSelector, BackreactionSelector	planned active	wahlfreier Selektor für UV-/Spektral- stufen soll später derived werden
BranchingSelector	Exportform reserviert; keine konkurrie-UVSpectralSelector rende Kurzform		selectedBranching, BackreactionSelector	planned active	explizite, später de-freie Auswahlre- gel für Verzweigungsdaten
BackreactionSelector	Exportform reserviert; keine konkurrie-BranchingSelector, selectedBranching rende Kurzform		BackreactionFixedPoint, EffectiveLambda	planned active	dokumentierte Auswahlregel für rückgekoppelte Skalenfestlegung
InfiniteCarrier	kein Exportalias; Lean-Name reserviert ComplementClosure		DerivedSpacetime, LocalCovariance	active extensi- on	explizite unendliche Tragerstruktur jenseits endlicher Approximanten
HorizonLevel	nur im angegebenen Namespace; keine RegularizationClosure, ParameterClosure Zweitbelegung der Symbolform		BackreactionFixedPoint, DerivedSpacetim e, EinsteinLimitSeed	active core	ersetzt rohes $L_{\max}$ durch abgeleitete IR-/Horizontskale
BackreactionFixedPoint	Exportform reserviert; keine konkurrie-HorizonLevel, Backreaction rende Kurzform		DerivedSpacetime, InterfaceCausality, Ei nsteinLimitSeed	active core	konsistente Rückkopplung von Entropie, Einfluss und Skale

Tabelle K.1-A3: Definitionskatalog der Kernobjekte – Objektbasis, Teil 3

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
ComplementLocalNet	structure	$\mathcal{A}_{\text{comp}}$	$\mathcal{A}_{\text{comp}}$	symbolic notation + text alias	$\mathcal{A}_{\text{comp}}$ und <i>Komplementnetz</i> verweisen auf <i>ComplementLocalNet</i>	CNNA/AQFT/ComplementLocalNet	CNNA/AQFT/Notation.lean	CNNA.AQFT	openscopedCNNA.AQFT	AQFT
InterfaceLocalNet	structure	$\mathcal{A}_{\text{int}}$	$\mathcal{A}_{\text{int}}$	symbolic notation + text alias	$\mathcal{A}_{\text{int}}$ und <i>Interfacenet</i> verweisen auf <i>InterfaceLocalNet</i>	CNNA/AQFT/InterfaceLocalNet	CNNA/AQFT/Notation.lean	CNNA.AQFT	openscopedCNNA.AQFT	AQFT
ComplementStateNet	structure	$\omega_{\text{comp}}$	$\omega_{\text{comp}}$	symbolic notation + text alias	$\omega_{\text{comp}}$ und <i>Komplementzustandsnetz</i> verweisen auf <i>ComplementStateNet</i>	CNNA/AQFT/ComplementStateNet	CNNA/AQFT/Notation.lean	CNNA.AQFT	openscopedCNNA.AQFT	AQFT
ComplementGNS	structure	<i>Komplement-GNS-Datum</i>	<i>CompGNS</i>	text alias	<i>Komplement-GNS-Datum</i> verweist auf <i>ComplementGNS</i> ; die Tupelform benennt dieselbe Datenlage und führt keinen zweiten Begriff ein	CNNA/AQFT/ComplementGNS	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	AQFT dCNNA.AQFT
SeparatingProperty	theorem/pro- perty	<i>zyklisch-separierende Lage</i>	<i>CycSep</i>	text alias	<i>zyklisch-separierende Lage</i> verweist auf <i>SeparatingProperty</i>	CNNA/AQFT/SeparatingProperty	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	AQFT dCNNA.AQFT
TomitaTakesakiData	structure	<i>Tomita-Takesaki-Datum</i>	<i>TTData</i>	text alias	<i>Tomita-Takesaki-Datum</i> verweist auf <i>TomitaTakesakiData</i> ; die Tupelform ist nur dieselbe Datenexposition	CNNA/AQFT/TomitaTakesakiData	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	AQFT dCNNA.AQFT
ComplementKMS	def/state	<i>Komplement-KMS-Zustand</i>	<i>CompKMS</i>	text alias	<i>Komplement-KMS-Zustand</i> verweist auf <i>ComplementKMS</i> ; $\omega_{\beta}$ ist nur im KMS-Kontext erlaubt	CNNA/AQFT/ComplementKMS	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	AQFT dCNNA.AQFT
ComplementSpectrumCondition	gate/pro- perty	<i>Komplement-Spektralbedingungstrum</i>	<i>CompSpec</i>	text alias	<i>Komplement-Spektralbedingung</i> verweist auf <i>ComplementSpectrumCondition</i>	CNNA/AQFT/ComplementSpectrumCondition	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	B-D dCNNA.AQFT
GradedStarAlgebra	structure	<i>GradedStarAlgebra</i>	—	selfspeaking Namen	identisch mit dem Namen	CNNA/AQFT/GradedStarAlgebra	kein eigener Aliasexport	CNNA.AQFT	keine Exportaktivierung	AQFT
ModularConjugation	def/theorem layer	<i>Modularkonjugation</i>	<i>ModConj</i>	text alias	<i>Modularkonjugation</i> verweist auf <i>ModularConjugation</i>	CNNA/AQFT/ModularConjugation	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	AQFT dCNNA.AQFT
BisognanoWichmannSeed	seed	<i>BW-Seed</i>	<i>BWSeed</i>	text alias	<i>BW-Seed</i> verweist auf <i>BisognanoWichmannSeed</i>	CNNA/AQFT/BisognanoWichmannSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export <i>openscope</i>	AQFT dCNNA.AQFT

Tabelle K.1-B3: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 3

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
ComplementLocalNet	nur im angegebenen Namespace; keine Zweitbelegung der Symbolform	ComplementSectorFamily, GeneralizedDtN	ComplementStateNet, ComplementGNS, ComplementSpectrumCondition	active core	lokales Observablenetz des dunklen Sektors
InterfaceLocalNet	nur im angegebenen Namespace; keine Zweitbelegung der Symbolform	GeneralizedDtN, SectorSplit	InterfaceCausality, LocalCovariance	active core	lokales Netz der Glueing-, Defekt- und Entropieträger
ComplementStateNet	nur im angegebenen Namespace; keine Zweitbelegung der Symbolform	ComplementLocalNet, SectorSysEnv	GlobalStateCone, ComplementGNS, ComplementKMS	active core	Zustandsfamilie vor GNS- und KMS-Schicht
ComplementGNS	Exportform reserviert; keine konkurrierende Kurzform	ComplementStateNet, ComplementLocalNet	SeparatingProperty, TomitaTakesakiData, TimeReversalOnGNS	active core	Darstellungsart des modularen B-Kerns
SeparatingProperty	Exportform reserviert; keine konkurrierende Kurzform	ComplementGNS	TomitaTakesakiData, ModularConjugation	active core	Scharnier zwischen GNS und Tomita-Takesaki
TomitaTakesakiData	Exportform reserviert; keine konkurrierende Kurzform	ComplementGNS, SeparatingProperty	ModularConjugation, ComplementKMS	active core	modularer Operator-, Konjugations- und Tomita-Kern
ComplementKMS	Exportform reserviert; keine konkurrierende Kurzform	ComplementStateNet, TomitaTakesakiData	ComplementSpectrumCondition, BisognanoWichmannSeed	active core	thermische und modulare Schicht des Komplementnetzes
ComplementSpectrumCondition	Exportform reserviert; keine konkurrierende Kurzform	ComplementKMS, ComplementLocalNet	BisognanoWichmannSeed, NuclearitySeed, CPTGate	active core	Vorbedingung für BW-, Scattering- und Kramers-nahe Strukturen
GradedStarAlgebra	kein Exportalias; Lean-Name reserviert	ComplementLocalNet, ComplementGNS	FermionicStatisticsSeed, KramersGate	active extension	algebraischer Untergrund für fermionische und gradierte Sektoren
ModularConjugation	Exportform reserviert; keine konkurrierende Kurzform	TomitaTakesakiData, SeparatingProperty	BisognanoWichmannSeed, TimeReversalOnGNS, CPTGate	active extension	kontrolliert die konjugative Seite des modularen Kerns
BisognanoWichmannSeed	Exportform reserviert; keine konkurrierende Kurzform	ModularConjugation, ComplementSpectrumCondition	LocalCovariance, CPTGate	active extension	Brücke von modularer Gruppe zu Boost-/Wedge-Lesarten

**Tabelle K.1-A4: Definitionskatalog der Kernobjekte – Objektbasis, Teil 4**

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
NuclearitySeed	seed	<i>Nuclearity-Seed</i>	<i>NucSeed</i>	text alias	<i>Nuclearity-Seed</i> verweist auf <i>NuclearitySeed</i>	CNNA/AQFT/NuclearitySeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT
Nuclearity	gate/theorem	Nuclearity	—	selfspeaking	identisch mit dem Namen	CNNA/Gates/Nuclearity.lean	kein eigener Aliasexport	CNNA	keine Exportaktivierung	AQFT/Gates
DHRsectorsSeed	seed	<i>DHR-Sektoren-Seed</i>	<i>DHRSeed</i>	text alias	<i>DHR-Sektoren-Seed</i> verweist auf <i>DHRsectorsSeed</i>	CNNA/AQFT/DHRsectorsSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT
FieldAlgebraReconstructionSeed	seed	<i>Feldalgebra-Rekonstruktions-Seed</i>	<i>FieldReconstructionSeed</i>	text alias	<i>Feldalgebra-Rekonstruktions-Seed</i> verweist auf <i>FieldAlgebraReconstructionSeed</i>	CNNA/AQFT/FieldAlgebraReconstructionSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT
ChargeStatisticsCoherence	gate	<i>Ladungs-Statistik-Kohärenz</i>	<i>ChargeStatisticsCoh</i>	text alias	<i>Ladungs-Statistik-Kohärenz</i> verweist auf <i>ChargeStatisticsCoherence</i>	CNNA/Gates/ChargeStatisticsCoherence.lean	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT/Gates
ChargeReconstructionTheorem	theorem	<i>Ladungsrekonstruktionssatz</i>	<i>ChargeReconstructionTheorem</i>	text alias	<i>Ladungsrekonstruktionssatz</i> verweist auf <i>ChargeReconstructionTheorem</i>	CNNA/AQFT/ChargeReconstructionTheorem	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT
DerivedSpacetime	structure	<i>derived spacetime</i>	<i>DerSpacetime</i>	text alias	<i>derived spacetime</i> verweist auf <i>DerivedSpacetime</i>	CNNA/PillarD/DerivedSpacetime	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	Pillar D
InterfaceCausality	structure/gate	<i>Interface-Kausalität</i>	<i>IfcCausality</i>	text alias	<i>Interface-Kausalität</i> verweist auf <i>InterfaceCausality</i>	CNNA/PillarD/InterfaceCausality	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	Pillar D
LocalCovariance	structure/gate	<i>lokale Kovarianz</i>	<i>LocCovariance</i>	text alias	<i>lokale Kovarianz</i> verweist auf <i>LocalCovariance</i>	CNNA/PillarD/LocalCovariance	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	Pillar D
RelativeCauchyEvolutionSeed	seed	<i>RelativeCauchyEvolutionSeed</i>	—	selfspeaking	identisch mit dem Namen; kein kurzer Symbolalias	CNNA/PillarD/RelativeCauchyEvolutionSeed	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	Pillar D
DynamicalLocality	gate	<i>DynamicalLocality</i>	—	selfspeaking	identisch mit dem Namen	CNNA/Gates/DynamicalLocality.lean	kein eigener Aliasexport	CNNA	keine Exportaktivierung	D/Gates

Tabelle K.1-B4: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 4

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
NuclearitySeed	Exportform reserviert; keine konkurrierende Kurzform	ComplementSpectrumCondition	Nuclearity, HadamardStateSeed, LocalCovariance	active extension	Phase-Space-Vorzone jenseits blossen Split-Verhaltens
Nuclearity	kein Exportalias; Lean-Name reserviert	NuclearitySeed	DHRsectorsSeed, ChargeStatisticsCoherence	active extension	Phase-Space- und Dichteschicht vor DHR-/Gauge-Rekonstruktion
DHRsectorsSeed	Exportform reserviert; keine konkurrierende Kurzform	Nuclearity, ComplementSpectrumCondition	FieldAlgebraReconstructionSeed, ChargeStatisticsCoherence	active extension	lokalisierte Superselection-Sektoren des Observablenetztes
FieldAlgebraReconstructionSeed	Exportform reserviert; keine konkurrierende Kurzform	DHRsectorsSeed	ChargeReconstructionTheorem, CPTTheorem	active extension	DR-artiger Weg zu Feldalgebra und kompakter Eichgruppe
ChargeStatisticsCoherence	Exportform reserviert; keine konkurrierende Kurzform	DHRsectorsSeed, Nuclearity	ChargeReconstructionTheorem, SpinStatisticsTheorem	active extension	prüft Kohärenz von Ladung, Statistik und rekonstruiertem Gauge-Layer
ChargeReconstructionTheorem	Exportform reserviert; keine konkurrierende Kurzform	FieldAlgebraReconstructionSeed, ChargeStatisticsCoherence	CPTTheorem, GaugeSectorSeed	active extension	expliziter DHR/DR-Abschluss vor voller Gauge-/Matter-Lesart
DerivedSpacetime	Exportform reserviert; keine konkurrierende Kurzform	BackreactionFixedPoint, HorizonLevel	InterfaceCausality, LocalCovariance, EinsteinLimitSeed	active extension	emergenter geometrischer Träger jenseits blosser Interface-Logik
InterfaceCausality	Exportform reserviert; keine konkurrierende Kurzform	DerivedSpacetime, InterfaceLocalNet	LocalCovariance, PTBridgeSeed	active extension	kausale Glueing- und Schnittstellenstruktur
LocalCovariance	Exportform reserviert; keine konkurrierende Kurzform	InterfaceCausality, BisognanoWichmannSeed	RelativeCauchyEvolutionSeed, HadamardStateSeed, CPTGate	active extension	LCQFT-nahe Kovarianzbedingung auf emergentem Hintergrund
RelativeCauchyEvolutionSeed	kein Exportalias; Lean-Name reserviert	LocalCovariance	DynamicalLocality, DerivedStressTensorSeed	active extension	dynamische Reaktion auf lokale Hintergrundvariation
DynamicalLocality	kein Exportalias; Lean-Name reserviert	RelativeCauchyEvolutionSeed	HadamardStateSeed, LocalWickPolynomialsSeed	active extension	Ableich von kinematischer und dynamischer Lokalität

Tabelle K.1-A5: Definitionskatalog der Kernobjekte – Objektbasis, Teil 5

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
HadamardStateSeed	seed	<i>Hadamard-State-Seed</i>	<i>HadSeed</i>	text alias	<i>Hadamard-State-Seed</i> verweist auf HadamardStateSeed	CNNA/PillarD/HadamardStateSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	Pillar D
MicrolocalSpectrumGate	gate	<i>mikrolokales Spektralgate</i>	<i>MicroSpec</i>	text alias	<i>mikrolokales Spektralgate</i> verweist auf MicrolocalSpectrumGate	CNNA/PillarD/MicrolocalSpectrumGate	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	Pillar D
LocalWickPolynomialsSeed	seed	<i>lokale-Wick-Polynome-Seed</i>	<i>WickSeed</i>	text alias	<i>lokale-Wick-Polynome-Seed</i> verweist auf LocalWickPolynomialsSeed	CNNA/PillarD/LocalWickPolynomialsSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	Pillar D
DerivedStressTensorSeed	seed	<i>DerivedStressTensorSeed</i>	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/PillarD/DerivedStressTensorSeed	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	Pillar D
EntanglementEquilibriumSeed	seed	<i>EntanglementEquilibriumSeed</i>	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/PillarD	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	Pillar D
EinsteinLimitSeed	seed	<i>Einstein-Limes-Seed</i>	<i>Einstein-Seed</i>	text alias	<i>Einstein-Limes-Seed</i> verweist auf EinsteinLimitSeed	CNNA/PillarD	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export openscope	D
InducedGravitySeed	seed	<i>InducedGravitySeed</i>	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/PillarD	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	D
TimeReversalOnGNS	def	<i>Zeitumkehr auf TRonGNS</i>	<i>TRonGNS</i>	text alias	<i>Zeitumkehr auf GNS</i> verweist auf TimeReversalOnGNS	CNNA/AQFT/TimeReversalOnGNS	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT/Symmetrie
AntiunitarySeed	seed	<i>AntiunitarySeed</i>	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/AQFT/AntiunitarySeed	kein eigener Aliasexport	CNNA.AQFT	keine Exportaktivierung	AQFT/Symmetrie
SpinRepresentationSeed	seed	<i>Spinrepräsentations-Seed</i>	<i>SpinSeed</i>	text alias	<i>Spinrepräsentations-Seed</i> verweist auf SpinRepresentationSeed	CNNA/AQFT/SpinRepresentationSeed	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope	AQFT/Symmetrie
FermionicStatisticsSeed	seed	<i>FermionicStatisticsSeed</i>	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/AQFT/FermionicStatisticsSeed	kein eigener Aliasexport	CNNA.AQFT	keine Exportaktivierung	AQFT/Symmetrie

Tabelle K.1-B5: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 5

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
HadamardStateSeed	Exportform reserviert; keine konkurrierende Kurzform	LocalCovariance, NuclearitySeed	MicrolocalSpectrumGate, LocalWickPolynomialsSeed	active extension	mikrolokal zulässige Zustandsklasse auf emergentem Hintergrund
MicrolocalSpectrumGate	Exportform reserviert; keine konkurrierende Kurzform	HadamardStateSeed	LocalWickPolynomialsSeed, DerivedStressTensorSeed	active extension	prüft Hadamard-/mikrolokale Zulässigkeit
LocalWickPolynomialsSeed	Exportform reserviert; keine konkurrierende Kurzform	HadamardStateSeed, MicrolocalSpectrumGate	DerivedStressTensorSeed, EinsteinLimitSeed	active extension	lokale kovariante Wick- und Zeitordnungsprodukte
DerivedStressTensorSeed	kein Exportalias; Lean-Name reserviert	LocalWickPolynomialsSeed, RelativeCauchyEvolutionSeed	EntanglementEquilibriumSeed, EinsteinLimitSeed, InducedGravitySeed	active extension	renormierter Energie-Impuls-Ausgang der lokalen Observablen
EntanglementEquilibriumSeed	kein Exportalias; Lean-Name reserviert	DerivedStressTensorSeed	EinsteinLimitSeed, InducedGravitySeed	planned active	Jacobson-nahe Route von Entropiegleichgewicht zu Gravitation
EinsteinLimitSeed	Exportform reserviert; keine konkurrierende Kurzform	DerivedStressTensorSeed, EntanglementEquilibriumSeed	InducedGravitySeed, SpectralTripleSeed	planned active	später Limes vom derived D-Pfad zu GR-artiger Dynamik
InducedGravitySeed	kein Exportalias; Lean-Name reserviert	EinsteinLimitSeed	SpectralTripleSeed, GaugeSectorSeed	planned active	alternative späte Route zu gravitativem Verhalten über Materie-/Zustandsbeitrag
TimeReversalOnGNS	Exportform reserviert; keine konkurrierende Kurzform	ComplementGNS, ModularConjugation	AntiunitarySeed, KramersGate	active extension	hebt die observablenseitige Zeitumkehr auf den Darstellungsraum
AntiunitarySeed	kein Exportalias; Lean-Name reserviert	TimeReversalOnGNS	KramersGate, CPTGate	active extension	formalisiert Antiunitarität der implementierten Zeitumkehr
SpinRepresentationSeed	Exportform reserviert; keine konkurrierende Kurzform	GradedStarAlgebra	FermionicStatisticsSeed, SpinStatisticsTheorem	active extension	Spin-/double-cover-Schicht vor fermionischer Statistik
FermionicStatisticsSeed	kein Exportalias; Lean-Name reserviert	SpinRepresentationSeed, GradedStarAlgebra	StatisticsGate, KramersGate	active extension	trennt fermionische von bosonischen Sektoren

Tabelle K.1-A6: Definitionskatalog der Kernobjekte – Objektbasis, Teil 6

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
StatisticsGate	gate	<i>Statistik-Gate</i>	<i>StatGate</i>	text alias	<i>Statistik-Gate</i> verweist auf StatisticsGate	CNNA/Gates/StatisticsGate.lean	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope dCNNA.AQFT	AQFT/Gates
SpinStatisticsTheorem	theorem	<i>Spin-Statistik-Theorem</i>	<i>Spin-StatThm</i>	text alias	<i>Spin-Statistik-Theorem</i> verweist auf SpinStatisticsTheorem	CNNA/AQFT/SpinStatisticsTheorem	CNNA/AQFT/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope dCNNA.AQFT	AQFT
PTBridgeSeed	seed	<i>PT-Bridge-Seed</i>	<i>PTSeed</i>	text alias	<i>PT-Bridge-Seed</i> verweist auf PTBridgeSeed	CNNA/PillarE/PT-BridgeSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope dCNNA.PillarE	D-E
ChargeConjugationSeed	seed	<i>CS</i> <i>Ladungskonjugations-Seed</i>	<i>CSeed</i>	text alias	<i>Ladungskonjugations-Seed</i> verweist auf ChargeConjugationSeed	CNNA/PillarE/ChargeConjugationSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope dCNNA.PillarE	E
CPTGate	gate	<i>CPT-Gate</i>	<i>CPTCheck</i>	text alias	<i>CPT-Gate</i> verweist auf CPTGate	CNNA/Gates/CPTGate.lean	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope dCNNA.PillarE	B-D-E
CPTTheorem	theorem	<i>CPT-Theorem</i>	<i>CPTThm</i>	text alias	<i>CPT-Theorem</i> verweist auf CPTTheorem	CNNA/AQFT/CPTTheorem	CNNA/PillarE/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope dCNNA.AQFT	B-D-E
KramersGate	gate	<i>Kramers-Gate</i>	<i>KramersCheck</i>	text alias	<i>Kramers-Gate</i> verweist auf KramersGate	CNNA/Gates/KramersGate.lean	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope dCNNA.PillarE	AQFT/E-Symmetrie
KramersTheorem	theorem	<i>Kramers-Theorem</i>	<i>KramersThm</i>	text alias	<i>Kramers-Theorem</i> verweist auf KramersTheorem	CNNA/AQFT/KramersTheorem	CNNA/PillarE/Notation.lean	CNNA.AQFT	kontextgebunden; bei Export openscope dCNNA.AQFT	AQFT/E-Symmetrie
GaugeSectorSeed	seed	<i>Gauge-Sektor-Seed</i>	<i>GaugeSeed</i>	text alias	<i>Gauge-Sektor-Seed</i> verweist auf GaugeSectorSeed	CNNA/PillarE/GaugeSectorSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope dCNNA.PillarE	E
AnomalyInflowSeed	seed	<i>Anomaly-Inflow-Seed</i>	<i>InflowSeed</i>	text alias	<i>Anomaly-Inflow-Seed</i> verweist auf AnomalyInflowSeed	CNNA/PillarE/AnomalyInflowSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope dCNNA.PillarE	E
AnomalyInflow	gate	AnomalyInflow	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/Gates/AnomalyInflow.lean	kein eigener Aliasexport	CNNA	keine Exportaktivierung	E/Gates

Tabelle K.1-B6: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 6

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
StatisticsGate	Exportform reserviert; keine konkurrie- rende Kurzform	FermionicStatisticsSeed	SpinStatisticsTheorem, KramersTheorem	active extension	Spin-Statistik-Konsistenztest auf gradierter Algebra
SpinStatisticsTheorem	Exportform reserviert; keine konkurrie- rende Kurzform	StatisticsGate, ChargeStatisticsCoherence	CPTTheorem, KramersTheorem	active extension	Endtheorem des Statistikstrangs
PTBridgeSeed	Exportform reserviert; keine konkurrie- rende Kurzform	InterfaceCausality	ChargeConjugationSeed, CPTGate	active extension	koppelt Raumspiegelungsdaten an die B-Zeitumkehrschicht
ChargeConjugationSeed	Exportform reserviert; keine konkurrie- rende Kurzform	PTBridgeSeed	CPTGate, CPTTheorem	active extension	Teilchen/Antiteilchen- bzw. C-Layer
CPTGate	Exportform reserviert; keine konkurrie- rende Kurzform	ChargeConjugationSeed, AntiunitarySeed	CPTTheorem, KramersGate	active extension	pfeilerübergreifende Abnahme von PT, C und CPT
CPTTheorem	Exportform reserviert; keine konkurrie- rende Kurzform	CPTGate, SpinStatisticsTheorem	KramersTheorem, GaugeSectorSeed	active extension	explizites Endtheorem des pfeilerübergreifenden CPT-Strangs
KramersGate	Exportform reserviert; keine konkurrie- rende Kurzform	TimeReversalOnGNS, CPTGate	KramersTheorem, ChiralMatterSeed	active extension	trennt $\Theta^2 = (-1)^F$ , Zeitumkehrkovarianz, Spektrum und Entartung
KramersTheorem	Exportform reserviert; keine konkurrie- rende Kurzform	KramersGate, StatisticsGate	GaugeSectorSeed, ChiralMatterSeed	active extension	explizites Endtheorem des graduierten Zeitumkehrsektors
GaugeSectorSeed	Exportform reserviert; keine konkurrie- rende Kurzform	CPTTheorem, KramersTheorem	AnomalyInflowSeed, ChargeQuantizationGate, HiggsSectorSeed	active extension	emergente Eichsektoren aus Interface, Defekt und Glueing
AnomalyInflowSeed	Exportform reserviert; keine konkurrie- rende Kurzform	GaugeSectorSeed	AnomalyInflow, AnomalyCancellationGate	active extension	Interface-/Boundary-Inflow-Schicht für chirale Materie
AnomalyInflow	kein Exportalias; Lean-Name reserviert	AnomalyInflowSeed	AnomalyCancellationGate, ChargeQuantizationGate	active extension	prüft die Tragung der chiralen Anomalielast durch Inflow

Tabelle K.1-A7: Definitionskatalog der Kernobjekte – Objektbasis, Teil 7

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
ChiralMatterSeed	seed	ChiralMatterSeed	—	selfspeaking	identisch mit dem Namen	Lean-CNNA/PillarE/ChiralMatterSeed	kein eigener Aliasexport	CNNA.PillarE	keine Exportaktivierung	E
AnomalyCancellationGate	gate	Anomalie-Kompensations-Gate	AnomCancellationGate	text alias	Anomalie-Kompensations-Gate verweist auf AnomalyCancellationGate	CNNA/Gates/AnomalyCancellationGateLean	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E/Gates
ChargeQuantizationGate	gate	Ladungsquantisierungs-Gate	ChargeQuantizationGate	text alias	Ladungsquantisierungs-Gate verweist auf ChargeQuantizationGate	CNNA/Gates/ChargeQuantizationGateLean	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E/Gates
FiniteInternalGeometrySeed	seed	FiniteInternalGeometrySeed	—	selfspeaking	identisch mit dem Namen	Lean-CNNA/PillarE/FiniteInternalGeometrySeed	kein eigener Aliasexport	CNNA.PillarE	keine Exportaktivierung	D-E
HiggsSectorSeed	seed	Higgs-Sektor-Seed	HiggsSeed	text alias	Higgs-Sektor-Seed verweist auf HiggsSectorSeed	CNNA/PillarE/HiggsSectorSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E
EWSBSeed	seed	EWSB-Seed	EWSBSeed	text alias	EWSB-Seed verweist auf EWSBSeed	CNNA/PillarE/EWSBSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E
YukawaFlavorSeed	seed	Yukawa-Flavor-Seed	YukawaSeed	text alias	Yukawa-Flavor-Seed verweist auf YukawaFlavorSeed	CNNA/PillarE/YukawaFlavorSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E
GenerationSeed	seed	Generationen-Seed	GenSeed	text alias	Generationen-Seed verweist auf GenerationSeed	CNNA/PillarE/GenerationSeed	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E
SMMatchingWindow	window	SM-Matching-Window	SMMatchWindow	text alias	SM-Matching-Window verweist auf SMMatchingWindow	CNNA/PillarE/SMMatchingWindow	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E
SMEFTWindow	window	SMEFT-Window	SMEFTWindow	text alias	SMEFT-Window verweist auf SMEFTWindow	CNNA/PillarE/SMEFTWindow	CNNA/PillarE/Notation.lean	CNNA.PillarE	kontextgebunden; bei Export openscope	E
SpectralTripleSeed	seed	SpectralTripleSeed	—	selfspeaking	identisch mit dem Namen	Lean-CNNA/PillarD/SpectralTripleSeed	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	D

Tabelle K.1-B7: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 7

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
ChiralMatterSeed	kein Exportalias; Lean-Name reserviert	KramersTheorem, GaugeSectorSeed	AnomalyCancellationGate, YukawaFlavorSeed	active extension	chirale Materie unter Doubling- und Inflow-Kontrolle
AnomalyCancellationGate	Exportform reserviert; keine konkurrierende Kurzform	AnomalyInflow, ChiralMatterSeed	ChargeQuantizationGate, SMEFTWindow	active extension	Abnahme vor belastbarer Gauge-/Chiralitätsrede
ChargeQuantizationGate	Exportform reserviert; keine konkurrierende Kurzform	AnomalyCancellationGate, GaugeSectorSeed	FiniteInternalGeometrySeed, SMMatchingWindow	active extension	prüft Ladungs- und Quantisierungsstruktur vor Higgs/Flavor
FiniteInternalGeometrySeed	kein Exportalias; Lean-Name reserviert	ChargeQuantizationGate	SpectralTripleSeed, DiracOperatorSeed	active extension	E-nahe Fortsetzung der NCG-Route für endliche interne Geometrie
HiggsSectorSeed	Exportform reserviert; keine konkurrierende Kurzform	GaugeSectorSeed	EWSBSeed, YukawaFlavorSeed	active extension	symmetry-breaking-Vorzone nach stabiler Gauge-/Chiralitätsschicht
EWSBSeed	Exportform reserviert; keine konkurrierende Kurzform	HiggsSectorSeed	YukawaFlavorSeed, GenerationSeed	active extension	elektroschwache Brechung als eigener abgeleiteter Strang
YukawaFlavorSeed	Exportform reserviert; keine konkurrierende Kurzform	EWSBSeed, ChiralMatterSeed	GenerationSeed, SMMatchingWindow	active extension	Flavor- und Yukawa-Strang
GenerationSeed	Exportform reserviert; keine konkurrierende Kurzform	YukawaFlavorSeed	SMMatchingWindow, SMEFTWindow	active extension	später Parallelstrang für Generationen- und Mischungsstruktur
SMMatchingWindow	Exportform reserviert; keine konkurrierende Kurzform	GenerationSeed, ChargeQuantizationGate	SMEFTWindow, DiracOperatorSeed	late window	spätes IR-Matching auf SM-artige Sektoren
SMEFTWindow	Exportform reserviert; keine konkurrierende Kurzform	SMMatchingWindow, AnomalyCancellationGate	TheorySpaceSeed	late window	spätes EFT-Fenster ausserhalb des Kerns
SpectralTripleSeed	kein Exportalias; Lean-Name reserviert	FiniteInternalGeometrySeed, EinsteinLimitSeed	DiracOperatorSeed, RealStructureSeed, SpectralActionSeed	active extension	spektrogeometrischer Kern für die NCG-Route

Tabelle K.1-A8: Definitionskatalog der Kernobjekte – Objektbasis, Teil 8

Objekt	Klasse	Primärform	Exportform	Modus	Expansion / Bindung	Herkunft	Zielmodul	Namespace	Aktivierungsregel	Kontext
DiracOperatorSeed	seed	<i>Dirac-Operator-Seed</i>	<i>DiracSeed</i>	text alias	<i>Dirac-Operator-Seed</i> verweist auf <code>DiracOperatorSeed</code>	CNNA/PillarD/DiracOperatorSeed	CNNA/PillarD/Notation.lean	CNNA.PillarD	kontextgebunden; bei Export <code>openscope dCNNA.PillarD</code>	D
RealStructureSeed	seed	RealStructureSeed	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/PillarD/RealStructureSeed	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	D
SpectralActionSeed	seed	SpectralActionSeed	—	selfspeaking	identisch mit dem Lean-Namen	CNNA/PillarD/SpectralActionSeed	kein eigener Aliasexport	CNNA.PillarD	keine Exportaktivierung	D-E
SpectralDimensionFlow	def	SpectralDimensionFlow	—	selfspeaking	identisch mit dem Lean-Namen	diagnostischer D-Pfad	kein eigener Aliasexport	CNNA	keine Exportaktivierung	D/Diagnostik
SpectralDimensionFlowSeed	seed	SpectralDimensionFlowSeed	—	selfspeaking	identisch mit dem Lean-Namen	diagnostischer D-Pfad	kein eigener Aliasexport	CNNA	keine Exportaktivierung	D/Diagnostik
CutoffModeCountingSeed	seed	CutoffModeCountingSeed	—	selfspeaking	identisch mit dem Lean-Namen	diagnostischer D-Pfad	kein eigener Aliasexport	CNNA	keine Exportaktivierung	D/Diagnostik
RunningBoundaryDataSeed	seed	RunningBoundaryDataSeed	—	selfspeaking	identisch mit dem Lean-Namen	diagnostischer D-Pfad	kein eigener Aliasexport	CNNA	keine Exportaktivierung	D/Diagnostik
TheorySpaceSeed	seed	TheorySpaceSeed	—	selfspeaking	identisch mit dem Lean-Namen	D/AS-Route	kein eigener Aliasexport	CNNA	keine Exportaktivierung	D/AS

Tabelle K.1-B8: Definitionskatalog der Kernobjekte – Kollisionen, Abhängigkeiten und Rollen, Teil 8

Objekt	Kollisionspolitik	logische Vorgänger	Verbraucher / Nachfolger	Status	Rolle
DiracOperatorSeed	Exportform reserviert; keine konkurrierende Kurzform	SpectralTripleSeed, FiniteInternalGeometrySeed	RealStructureSeed, SpectralActionSeed	active extension	abgeleitete Dirac-Struktur der spektrogeometrischen D-Linie
RealStructureSeed	kein Exportalias; Lean-Name reserviert	SpectralTripleSeed, DiracOperatorSeed	SpectralActionSeed, TheorySpaceSeed	active extension	KO- und Realstruktur der NCG-Brücke
SpectralActionSeed	kein Exportalias; Lean-Name reserviert	SpectralTripleSeed, RealStructureSeed	SpectralDimensionFlow, RunningBoundaryDataSeed	active extension	spektrogeometrische Wirkungsroute als Brücke nach E
SpectralDimensionFlow	kein Exportalias; Lean-Name reserviert	SpectralActionSeed	SpectralDimensionFlowSeed, TheorySpaceSeed	planned active	laufende spektrale Dimensionsdiagnostik
SpectralDimensionFlowSeed	kein Exportalias; Lean-Name reserviert	SpectralDimensionFlow	CutoffModeCountingSeed, RunningBoundaryDataSeed	planned active	Vorzone für spektrale Dimensionsflüsse
CutoffModeCountingSeed	kein Exportalias; Lean-Name reserviert	SpectralDimensionFlowSeed	RunningBoundaryDataSeed, TheorySpaceSeed	planned active	Modenzahl- und cutoffbezogene Spektraldiagnostik
RunningBoundaryDataSeed	kein Exportalias; Lean-Name reserviert	CutoffModeCountingSeed, SpectralActionSeed	TheorySpaceSeed, SMMatchingWindow	planned active	laufende Randdaten entlang Skalen- oder RG-artiger Entwicklung
TheorySpaceSeed	kein Exportalias; Lean-Name reserviert	RunningBoundaryDataSeed, SpectralDimensionFlow	SMMatchingWindow, SMEFTWindow	planned active	asymptotic-safety-nahe Theorie- und Flussstruktur

## L Quellenangaben

### Literatur

- [1] REAL-OQS project archive, *REAL-OQS\_v0.0605\_full\_no\_comments.zip* 2026-03-29.
- [2] Jan Seeck, *Complement-Netzarchitektur (CNNA) aus dem ToC*, internal project specification v0.1, uploaded PDF, 2026.
- [3] Jan Seeck, *Regelwerk für codebasierte Lean-Dokumentationen. Externe Dokumentation statt Code-Kommentierung*, Revision 1.8, uploaded PDF, 2026.
- [4] Mykola Dedushenko, *Snowmass White Paper: The Quest to Define QFT*, Int. J. Mod. Phys. A **38** (2023) 2330002, arXiv:2203.08053.
- [5] Rudolf Haag and Daniel Kastler, *An Algebraic Approach to Quantum Field Theory*, J. Math. Phys. **5** (1964), 848–861, DOI: 10.1063/1.1704187.
- [6] Romeo Brunetti, Klaus Fredenhagen, and Rainer Verch, *The Generally Covariant Locality Principle: A New Paradigm for Local Quantum Physics*, Commun. Math. Phys. **237** (2003), 31–68, arXiv:math-ph/0112041.
- [7] Christopher J. Fewster and Benjamin Lang, *Dynamical Locality of the Free Maxwell Field*, Ann. Henri Poincaré **17** (2016), 401–436, arXiv:1403.7083.
- [8] Marco Benini, Alexander Schenkel, and Lukas Woike, *Operads for Algebraic Quantum Field Theory*, Commun. Contemp. Math. **23** (2021) 2050007, arXiv:1709.08657.
- [9] Michl Atiyah, *Topological Quantum Field Theories*, Publ. Math. IHES **68** (1988), 175–186.
- [10] Gräme Segal, *The Definition of Conformal Field Theory*, in *Topology, Geometry and Quantum Field Theory*, London Mathematical Society Lecture Note Series, Cambridge University Press, 2004, pp. 421–577.
- [11] Jürgen Fuchs, Ingo Runkel, and Christoph Schweigert, *Boundaries, Defects and Frobenius Algebras*, arXiv:hep-th/0302200.
- [12] Jürg Fröhlich, Jürgen Fuchs, Ingo Runkel, and Christoph Schweigert, *Duality and Defects in Rational Conformal Field Theory*, Nucl. Phys. B **763** (2007), 354–430, arXiv:hep-th/0607247.
- [13] Ted Jacobson, *Thermodynamics of Spacetime: The Einstein Equation of State*, Phys. Rev. Lett. **75** (1995), 1260–1263, arXiv:gr-qc/9504004.
- [14] Martin Reuter, *Nonperturbative Evolution Equation for Quantum Gravity*, Phys. Rev. D **57** (1998), 971–985, arXiv:hep-th/9605030.
- [15] Martin Reuter and Frank Saueressig, *Quantum Einstein Gravity*, New J. Phys. **14** (2012) 055022, arXiv:1202.2274.
- [16] Oliver Lauscher and Martin Reuter, *Fractal Spacetime Structure in Asymptotically Safe Gravity*, JHEP **10** (2005) 050, arXiv:hep-th/0508202.
- [17] Matt Visser, *Sakharov’s Induced Gravity: A Modern Perspective*, Mod. Phys. Lett. A **17** (2002), 977–992, arXiv:gr-qc/0204062.
- [18] Romeo Brunetti, Klaus Fredenhagen, and Kasia Rejzner, *Locally Covariant Approach to Effective Quantum Gravity*, arXiv:2212.07800.

- [19] Ali H. Chamseddine and Alain Connes, *The Spectral Action Principle*, arXiv:hep-th/9606001.
- [20] Ali H. Chamseddine and Alain Connes, *Why the Standard Model*, arXiv:0706.3688.
- [21] Holger Bech Nielsen and Masao Ninomiya, *A No-Go Theorem for Regularizing Chiral Fermions*, Phys. Lett. B **105** (1981), 219–223, DOI: 10.1016/0370-2693(81)91026-1.
- [22] Curtis G. Callan Jr. and Jeffrey A. Harvey, *Anomalies and Fermion Zero Modes on Strings and Domain Walls*, Nucl. Phys. B **250** (1985), 427–436, DOI: 10.1016/0550-3213(85)90489-4.
- [23] Luis Alvarez-Gaume and Migül Angel Vazquez-Mozo, *Anomalies and the Green-Schwarz Mechanism*, arXiv:2211.06467.
- [24] Sergio Doplicher, Rudolf Haag, and John E. Roberts, *Local Observables and Particle Statistics I*, Commun. Math. Phys. **23** (1971), 199–230, DOI: 10.1007/BF01877742.
- [25] Sergio Doplicher and John E. Roberts, *Why There is a Field Algebra with a Compact Gauge Group Describing the Superselection Structure in Particle Physics*, Commun. Math. Phys. **131** (1990), 51–107, DOI: 10.1007/BF02097680.
- [26] Detlev Buchholz and Eyvind H. Wichmann, *Causal Independence and the Energy-Level Density of States in Local Quantum Field Theory*, Commun. Math. Phys. **106** (1986), 321–344, DOI: 10.1007/BF01454978.
- [27] Detlev Buchholz and John E. Roberts, *New Light on Infrared Problems: Sectors, Statistics, Symmetries and Spectrum*, Commun. Math. Phys. **330** (2014), 935–972, arXiv:1304.2794.
- [28] Detlev Buchholz, *Scattering in Relativistic Quantum Field Theory: Basic Concepts, Tools, and Results*, arXiv:math-ph/0509047.
- [29] H. J. Borchers, *The CPT-Theorem in Two-Dimensional Theories of Local Observables*, Commun. Math. Phys. **143** (1992), 315–332, DOI: 10.1007/BF02099011.
- [30] Rudolf Haag, *Local Quantum Physics: Fields, Particles, Algebras*, 2nd rev. ed., Texts and Monographs in Physics, Springer, Berlin, Heidelberg, 1996, DOI: 10.1007/978-3-642-61458-3.
- [31] Wojciech Dybalski, *Haag-Ruelle Scattering Theory in Presence of Massless Particles*, Lett. Math. Phys. **72** (2005), 27–38, arXiv:hep-th/0412226.
- [32] Christopher J. Fewster and Rainer Verch, *Dynamical Locality and Covariance: What Makes a Physical Theory the Same in All Spacetimes?*, Ann. Henri Poincaré **13** (2012), 1613–1674, DOI: 10.1007/s00023-012-0165-0.
- [33] Marek J. Radzikowski, *Micro-Local Approach to the Hadamard Condition in Quantum Field Theory on Curved Space-Time*, Commun. Math. Phys. **179** (1996), 529–553, DOI: 10.1007/BF02100096.
- [34] Romeo Brunetti, Klaus Fredenhagen, and Martin Köhler, *The Microlocal Spectrum Condition and Wick Polynomials of Free Fields on Curved Spacetimes*, Commun. Math. Phys. **180** (1996), 633–652, DOI: 10.1007/BF02099626.
- [35] Stefan Hollands and Robert M. Wald, *Local Wick Polynomials and Time Ordered Products of Quantum Fields in Curved Spacetime*, Commun. Math. Phys. **223** (2001), 289–326, DOI: 10.1007/s002200100540.
- [36] Stefan Hollands and Robert M. Wald, *Existence of Local Covariant Time Ordered Products of Quantum Fields in Curved Spacetime*, Commun. Math. Phys. **231** (2002), 309–345, DOI: 10.1007/s00220-002-0719-y.

- 
- [37] Helmut Reeh and Siegfried Schlieder, *Bemerkungen zur Unitäräquivalenz von Lorentzinvarianten Feldern*, Nuovo Cimento **22** (1961), 1051–1068, DOI: 10.1007/BF02787889.
- [38] Masamichi Takesaki, *Tomita's Theory of Modular Hilbert Algebras and its Applications*, Lecture Notes in Mathematics **128**, Springer, 1970, DOI: 10.1007/BFb0065832.
- [39] Joseph J. Bisognano and Eyvind H. Wichmann, *On the Duality Condition for a Hermitian Scalar Field*, J. Math. Phys. **16** (1975), 985–1007, DOI: 10.1063/1.522605.
- [40] Joseph J. Bisognano and Eyvind H. Wichmann, *On the Duality Condition for Quantum Fields*, J. Math. Phys. **17** (1976), 303–321, DOI: 10.1063/1.522898.
- [41] Wolfgang Pauli, *The Connection Between Spin and Statistics*, Phys. Rev. **58** (1940), 716–722, DOI: 10.1103/PhysRev.58.716.
- [42] Daniele Guido and Roberto Longo, *An Algebraic Spin and Statistics Theorem*, Commun. Math. Phys. **172** (1995), 517–533, DOI: 10.1007/BF02101806.
- [43] Gerhart Lüders, *Proof of the TCP Theorem*, Ann. Phys. **2** (1957), 1–15, DOI: 10.1016/0003-4916(57)90032-5.
- [44] Richard P. Feynman, *The Reason for Antiparticles*, in *Paul Dirac: The Man and His Work*, Cambridge University Press, published online 2014, DOI: 10.1017/CBO9781107590076.002.
- [45] Particle Data Group, *Review of Particle Physics: The Standard Model and Electroweak Model and Constraints on New Physics*, PDG review article, 2023 edition.
- [46] Gino Isidori, Felix Wilsch, and Daniel Wyler, *The Standard Model Effective Field Theory at Work*, Rev. Mod. Phys. **96** (2024) 015006, arXiv:2303.16922.