



E N K I D U

— M I O V S K I G R O U P —

K O N Z E P T A R C H I T E K T U R

Helios Grid

Die Steuerungsebene der verteilten Intelligenz — Modelle ausrollen, Gesundheit überwachen und auf Ereignisse reagieren über Tausende von Knoten, wobei nur Telemetrie und Signal aufwärts fließen, niemals Rohdaten. Strukturiert durch Tangram, gesteuert durch Origi.

Deutsch · Kompaktausgabe

Nach Version 0.1 · 4. Juli 2026

Erstellt von MIOVSKI GROUP

Einstufung · Nicht klassifiziert · Entwurf zur internen Prüfung

Die vollständige englische Ausgabe ist maßgeblich

Zu dieser Ausgabe

Dieses Dokument ist eine deutsche Kompaktausgabe von « Helios Grid — The control plane for distributed intelligence », Version 0.1 (4. Juli 2026), erstellt von der Miovski Group. Jeder Abschnitt des Originals ist in gekürzter Form vertreten. Bei Abweichungen ist die vollständige englische Ausgabe maßgeblich. Einstufung: nicht klassifiziert — Entwurf zur internen Prüfung.

Zusammenfassung

Ein verteiltes Intelligenzgewebe der Northwind-Klasse steht und fällt mit einem Problem, das weder Sensorik noch Inferenz ist: dem Betrieb der Flotte. Hunderte bis Tausende Edge-Knoten — von Trupps getragen, auf Fahrzeugen montiert, in autonome Plattformen eingebettet oder an Masten geschraubt — müssen enrolliert, konfiguriert, gesund gehalten, mit neuen Modellen und Richtlinien aktualisiert und bei Ausfall wiederhergestellt werden, über intermittierende, umkämpfte und bandbreitenarme Verbindungen. Von Hand skaliert das nicht über eine Handvoll Knoten hinaus; mit einem konventionellen Cloud-Geräteverwaltungsdienst verletzt es die Souveränitäts-, Einstufungs- und Trennungsaufgaben, unter denen diese Gewebe leben.

Helios Grid ist Enkidus Antwort: eine souveräne, selbst gehostete Steuerungsebene für verteilte Intelligenz und das dritte Grundframework des Enkidu-Stacks. Wo Tangram einem Gewebe seine operative Struktur gibt (Zonen, Kontinuen, vertragliche Übergaben) und Origi sein Laufzeitvertrauen (Richtliniendurchsetzung, Vertrauens-Operationszonen, Beweiskette), gibt Helios ihm seine Betreibbarkeit: Flottenlebenszyklus, gestaffelte Modell- und Software-Rollouts, Gesundheitsbeobachtung und automatisierte Ereignisreaktion. Seine prägende Disziplin ist das Prinzip Signal statt Daten": Helios sendet nur Telemetrie und Signal aufwärts, niemals Rohdaten — Sichtbarkeit im Flottenmaßstab ohne Bandbreite im Flottenmaßstab, und ohne je einen zweiten Pfad zu schaffen, über den Missionsdaten einen Knoten verlassen könnten.

Helios folgt der über die Enkidu-Frameworks etablierten Prämisse — Determinismus, wo es zählt, Schlussfolgern, wo es hilft — und ergänzt sein eigenes Korollar: Deklaration, wo sie skaliert. Der Flottenzustand wird deklariert, signiert und per Pull abgeglichen, sodass ein Knoten, der einen Tag im Dunkeln war, zum selben Zustand konvergiert wie ein Knoten, der seine Verbindung nie verlor. Die Architektur übernimmt bewusst die bewährtesten Muster des NVIDIA-Edge-KI-Ökosystems — Fleet Commands Flottenorchestrierung, die Basisdienste der Jetson Platform Services, ausfallsichere A/B-Updates, Tritons Live-Modellverwaltung, GPU-Telemetrie der DCGM-Klasse, Mission Controls autonome Wiederherstellung, GitOps-Pull-Abgleich — und fasst sie hinter einer einzigen, gesteuerten, souveränen Steuerungsebene zusammen, die Origi überwacht und Tangram strukturiert.

1. Zweck und Geltungsbereich

Dies ist eine Konzeptarchitektur, kein Implementierungshandbuch: Sie definiert Helios Grids Struktur, seine tragenden Entscheidungen und seine Verträge mit den anderen Enkidu-Frameworks. Die Referenzimplementierung ist Northwind — die härteste Version des Problems: vier Rechenebenen, umkämpfte Verbindungen, Sicherheitseinstufung, autonome Wirte. Dieselbe Architektur, mit zivilen Bindungen, ist die Steuerungsebene für Lamassu (urbane Sensorik) und ortsfeste Installationen wie intelligente Gesundheitseinheiten. Eine Steuerungsebene, viele Gewebe.

Was Helios nicht ist: weder Datenplattform noch Fusionsmaschine noch Vertrauensframework. Es nimmt niemals Sensortelemetrie auf, trägt niemals Beobachtungen oder Bewertungen, fällt niemals ein Missionsurteil und erweitert niemals die Autorität eines Agenten. Seine Weigerung, Missionsdaten zu berühren, ist keine Einschränkung; sie ist das Design.

- Flottenlebenszyklus im Maßstab — berührungsloses Enrollment, Knoten-Zwillinge, deklarierter Sollzustand, Pull-Abgleich.
- Sichere Artefaktverteilung — signierte, gestaffelte, gesundheitsgeprüfte Rollouts mit A/B-Installation und automatischem Rollback.
- Beobachtbarkeit im Flottenmaßstab — Gesundheits- und Leistungssicht auf jeder Ebene, innerhalb einer strikten Telemetrie-Hülle.
- Autonome Ereignisreaktion — Erkennung, Klassifizierung und begrenzte automatisierte Behebung, mit menschlicher Eskalation jenseits des Runbooks.

2. Entwurfsprämisse und Prinzipien

Helios ruht auf einem Satz: zentral deklarieren, lokal abgleichen, nach Ausnahme berichten. Das Zentrum deklariert, was die Flotte sein soll; jeder Knoten zieht in eigenem Takt auf diese Deklaration zu und übersteht jede Unterbrechung; und die Flotte meldet aufwärts nur das Signal, das ein Bediener braucht — Gesundheit, Abweichung und Ereignisse — niemals die Daten, die die Mission erzeugt.

- 01 · Steuerung trägt niemals Daten — keine Missionsnutzlast durchquert je einen Helios-Kanal; die Telemetrieschemata können Missionsinhalte strukturell nicht ausdrücken.
- 02 · Zustand wird deklariert und gezogen — der Sollzustand wird zentral signiert; Knoten gleichen per Pull ab; Trennung ist ein Normalzustand, kein Fehler.
- 03 · Jedes Artefakt wird signiert und zugelassen — Helios verteilt; Origi lässt zu. Eine ungültige Signatur oder gebrochene Herkunft stoppt die Installation am Knoten, gleich was das Zentrum befahl.
- 04 · Updates sind konstruktiv ausfallsicher — jede veränderliche Installation nutzt einen A/B-Mechanismus mit automatischem Rollback; ein Fehlschlag degradiert zum vorherigen Zustand, nie zu einem unbrauchbaren Knoten.
- 05 · Behebung ist begrenzt — automatisierte Reaktionen stammen aus einem signierten Runbook vorab genehmigter Aktionen mit expliziten Wirkradius-Grenzen; alles darüber hinaus eskaliert an einen Menschen.
- 06 · Die Steuerungsebene ist selbst gesteuert — Agent, Warden und Core sind Agenten unter Origi: Vertrauenszonen, Deny-by-Default-Richtlinien, Beweiskette. Der Betreiber der Flotte ist so rechenschaftspflichtig wie die Flotte.

Warum kein Cloud-Geräteverwaltungsdienst? Helios übernimmt, was dieses Muster bewiesen hat — eine Konsole, One-Touch-Provisionierung, Fernverwaltung des Lebenszyklus — verwirft aber sein Bereitstellungsmodell: Enkidu-Gewebe operieren unter Auflagen zu Datensouveränität, Einstufung (Protected B in Kanada) und getrenntem Betrieb, die eine fremde Cloud-Steuerungsebene nicht erfüllen kann. Helios ist daher am zentralen Tier des Gewebes selbst gehostet, erbt dessen Akkreditierungsgrenze und funktioniert bei völlig fehlendem WAN.

3. Position im Enkidu-Stack

- Tangram — Wo findet Arbeit statt, und unter welchen Randbedingungen? — Struktur.
- Origi — Darf diese Handlung geschehen, und können wir beweisen, was geschah? — Vertrauen.
- Helios — Ist das Gewebe selbst gesund, aktuell und wiederherstellbar? — Betrieb.

Die drei greifen ineinander, ohne sich zu überlappen: Helios ist durch Tangram strukturiert (Flottensegmente sind Zonen; eine Rollout-Welle ist eine Zonenoperation mit Ein- und Austrittsbedingungen; ein gescheiterter Rollout betritt eine konstruierte Fehlerzone); Helios ist durch Origi gesteuert (ein Artefakt zu verteilen ist ein Vorschlag; Origis Zulassung — Signatur, Herkunft, Richtlinienbindung — ist der Commit); und Origi wie Tangram werden von Helios betrieben (Richtlinienpakete und Zonendefinitionen sind selbst versionierte Artefakte). Kein Komponent darf zugleich entscheiden und gewähren: Helios beobachtet und führt aus; Origi urteilt; Tangram begrenzt. Eine kompromittierte Steuerungsebene kann schlimmstenfalls vorschlagen.

4. Referenztopologie — Helios auf dem Northwind-Gewebe

- FIB-Knoten (Orin-Klasse) — Helios Agent — kryptografische Identität und Enrollment; Abgleichschleife; Artefaktbezug, -verifikation und A/B-Installation; lokale Telemetrie; lokaler Runbook-Vollstrecker.
- FIB-Hub (Thor-Klasse) — Helios Warden — Sub-Steuerungsebene seines Clusters: Cache für Sollzustand und Artefakte; Telemetrie-Aggregation und -Ausdünnung; lokale Alarmregel-Auswertung; Wellenkoordination; Out-of-Band-Wiederherstellungshaken.
- FOB-IFC (GB300-Klasse) — Helios Core (vorn) — Register und Zwillinge im Theatermaßstab; Rollout-Orchestrierung über Dutzende Hubs; Telemetriesee und Ereignismaschine; vordere Flottenkonsole.
- Central IFC (NVL72-Klasse) — Helios Core (zentral) — Register der Flotte der Flotten; maßgeblicher Sollzustandsspeicher und Artefaktregister; Langzeittelemetrie; Integration in das Infrastrukturmanagement des Rechenzentrums.

Zwei Eigenschaften dominieren. Erstens ist der Hub die Einheit der Autonomie: Ein Warden hält alles, was sein Cluster braucht, sodass ein vom FOB-IFC abgeschnittenes Cluster lokal weiter abgleicht, überwacht und behebt. Zweitens folgt die Verteilung dem Gewebe: Ein Modellartefakt reist pro Verbindung nur einmal — ein 2-GB-Update für ein 40-Knoten-Cluster kostet die Rückverbindung eine Übertragung, nicht vierzig.

5. Die drei Ebenen von Helios

- Zustandsebene — Knoten-Zwillinge, Sollzustandsspeicher, Artefaktregister. Deterministisch, versioniert, signiert: die einzige Quelle der Wahrheit.
- Flusebene — Abgleich, Verteilung, Telemetrietransport. Pull-basiert, wiederaufnehmbar, prioritätsklassiert, hüllenbegrenzt — ausgelegt auf die schlechteste Verbindung der Flotte, nicht die beste.
- Reaktionsebene — Ereignismaschine, Alarmregeln, Runbooks, Eskalationspfade. Begrenzte Automatisierung mit konstruierter menschlicher Naht.

Ein viertes Anliegen — Governance — ist bewusst keine Helios-Ebene: Es ist Origi, das alle drei umhüllt.

6. Flottenlebenszyklus-Management

Das Leben eines Knotens beginnt vor dem Einsatz: Bei der Provisionierung erhält das Gerät seine kryptografische Identität (hardwareverankert, wo die Ebene es erlaubt), sein Ebenenprofil und einen Basis-Cache des Sollzustands. Im Feld folgt der erste Start dem berührungslosen Enrollment; weil Sollzustand und Basismodelle auf dem Gerät mitreisen, ist ein Knoten nützlich, bevor er verbunden ist — und verwaltet, sobald er es ist. Das Enrollment erzeugt den Knoten-Zwilling — den maßgeblichen Datensatz, über den Bediener und Ereignismaschine schlussfolgern.

Der Abgleich ist idempotent und wiederaufnehmbar, nach dem GitOps-Modell: ein deklarerter, versionierter Sollzustand, den leichte Agenten ziehen und auf den sie konvergieren — das einzige Modell, das Northwinds Einsatzprofile übersteht, einschließlich MANET-Betrieb bei verweigerter Infrastruktur. Knoten verlassen die Flotte auf drei Wegen, alle konstruiert: geordneter Ruhestand (Schlüssel widerrufen, Zwilling archiviert, Artefakte gelöscht), Kompromittierungsverdacht (Origami-Quarantäne, Enrollment widerrufen, das Mesh angewiesen, ihn zu meiden, Beweise gesichert) und physischer Verlust (Ruheverschlüsselung und Schlüsselverwahrung begrenzen, was ein erbeuteter Knoten preisgeben kann). Eine Flotte, die nur das Hinzufügen von Knoten plant, ist eine halbe Flotte.

7. Artefakt- und Modellverteilung

Alles, was ein Knoten ausführt oder befolgt, ist ein Artefakt: feinabgestimmte, quantisierte Spezialistenmodelle (und ihre Adapter), Container-Images, Tangram-Zonendefinitionen, Origi-Richtlinienpakete, Ingestion-Konnektoren, Basissystem-Images. Das Artefaktregister am zentralen Core ist das souveräne Gegenstück eines privaten Registers: ein signierter Katalog, aus dem jedes einsetzbare Objekt — und nichts sonst — bezogen werden darf. Ein Modell, das seine Herkunft nicht zeigen kann, kommt nicht ins Register; ein Modell außerhalb des Registers erreicht keinen Knoten.

Kein Artefakt erreicht die ganze Flotte auf einmal: Ein Rollout deklariert seine Wellen — Kanarienvogel, Cluster, Theater — und die Gesundheitsschranken dazwischen; ein Schrankenversagen friert die Welle ein und löst automatisches Rollback bereits aktualisierter Knoten aus. Bei der Installation ist alles transaktional: volle A/B-Strategie für Systemartefakte (der Slot, der den Gesundheitscheck nicht besteht, wird nie festgeschrieben); für Modelle erlaubt die Steuer-API des Inferenzservers Laden, Entladen und Austausch im laufenden Betrieb, ohne Verfügbarkeitsverlust. Delta-Übertragungen halten beschränkte Verbindungen tragfähig; Hubs tragen zudem Out-of-Band-Wiederherstellungsfähigkeit — Hardware-Stromzyklen und Recovery-Mode-Reflash — um einen Knoten ohne Vor-Ort-Besuch wiederzubeleben.

8. Telemetrie, Gesundheit und das Prinzip Signal statt Daten"

Helios' Beobachtbarkeit ist rückwärts von seiner Bandbreiten- und seiner Vertrauensauflage konstruiert, die zusammenfallen: Die Edge reduziert; das Gewebe leitet Signal weiter; Rohdaten verlassen den Knoten niemals über einen Steuerkanal. Die Telemetrieschemata haben kein Feld, in dem ein Videobild, eine RF-Aufnahme oder eine Bewertung reisen könnte.

Jeder Knoten und Hub hält eine signierte Telemetrie-Hülle: Budget pro Intervall, Prioritätsklassen (Ereignisse > Gesundheitswerte > ausgedünnte Reihen > Logs auf Anfrage) und Degradationsverhalten. Jede Aufwärtsverbindung trägt eine Größenordnung weniger als die darunter. Der Warden berechnet einen Gesundheitswert je Knoten — eine Zahl, nach der ein Bediener tausend Knoten reihen kann — während die vollen Reihen auf der Ebene abfragbar bleiben, die sie hält; Alarmregeln laufen auf der niedrigsten Ebene, die die Daten hat, sodass ein Knoten seinen eigenen Alarm schlägt, selbst wenn er allein im Dunkeln ist. Modellgesundheit ist eine Metrikkategorie ersten Ranges: Konfidenzverteilungen, Enthaltungsraten, Driftindikatoren — der operative Zufluss für Origis Driftüberwachung und der Auslöser für Nachtrainingsanfragen.

9. Ereignisreaktion und automatisierte Behebung

Ereignisse — Alarme, Schrankenversagen, Enrollment-Anomalien, Origi-Vertrauensherabstufungen, Verbindungszustandswechsel — fließen in die Ereignismaschine jedes Cores, werden nach Schwere und Wirkradius klassifiziert, über die Flotte korreliert (ein schlechtes Artefakt sieht aus wie fünfzig gleichzeitige Knotenereignisse; die Maschine muss eine Ursache sehen) und gegen das Runbook geprüft. Ein Runbook-Eintrag ist eine signierte, versionierte, vorab genehmigte automatisierte Reaktion mit expliziten Vorbedingungen und Grenzen. Die Autonomie ist bewusst konservativ — neu starten, zurückrollen, isolieren, Last abwerfen, Wiederherstellung anfordern — und asymmetrisch in derselben Richtung wie Origis Vertrauenszonen: Automatisiertes darf die Aktivität oder Fähigkeit eines Knotens sofort verringern, aber nichts Automatisiertes erhöht je Fähigkeit oder Reichweite. Alles außerhalb des Runbooks eskaliert — mit Zwilling, korrelierter Ereigniskette und relevanter Telemetrie bereits in der Bedieneransicht versammelt.

10. Sicherheit und Governance — Helios unter Origi

Helios ist das wertvollste Ziel in jedem Gewebe, das es verwaltet: Es berührt jeden Knoten und verteilt ausführbare Artefakte. Seine Sicherheitshaltung ist daher nicht selbstbehauptet; es ist die von Origi, angewandt auf Helios wie auf jedes andere Agentensystem: jede Verbindung wechselseitig authentifiziert und verschlüsselt; Deny-by-Default-Autorität (kein Komponent kann die eigene Richtlinie erweitern); Zulassung über Verteilung (die Verifikation läuft am Knoten, im letzten Moment vor der Installation — ein kompromittierter Cache kann höchstens ein Artefakt liefern, das der Knoten ablehnt); Funktionstrennung (Bewertung, Urteil und Ausführung sind getrennte Komponenten, die die signierten Ausgaben der anderen konsumieren); vollständige Auditierbarkeit (Wer hat dieses Modell auf diesen Knoten gebracht, wann, unter wessen Autorität, und was hat es ersetzt?" ist stets aus der Spur beantwortbar); und Einstufungsdisziplin auf dem Steuerverkehr selbst.

11. Ausrichtung am NVIDIA-Werkzeugsatz

Helios erfindet nicht neu, was das NVIDIA-Edge-KI-Ökosystem bereits bewiesen hat; es übernimmt, härtet und hostet selbst: das Betriebsmuster von Fleet Command (unter Ablehnung seines Cloud-Modells), die Jetson Platform Services als knotenlokales Telemetriesubstrat, die validierte A/B-Strategie (Mender/L4T) als Installationsinvariante, Out-of-Band-Wiederherstellung der Allxon-Klasse, Triton-Modellserving im expliziten Steuermodus, die Modellproduktions-Pipeline nach TAO-Muster, die ausschließlich ins Helios-Register publiziert, DCGM-Telemetrie auf den Serverebenen, Mission Control als interne Verwaltungsebene des Central IFC und GitOps-Abgleich mit einer leichten Kubernetes-Distribution der K3s-Klasse auf Hub- und Zentralebene.

12. Skalierung und Fehlerverhalten

Das Skalenmodell folgt dem Gewebe: Dutzende Knoten je Warden, Dutzende Wardens je vorderem Core, föderierte Cores im Zentrum. Unter Verlust: Ein isolierter Knoten fühlt und inferiert weiter und gleicht bei Rückkehr der Verbindung ab; ein vom Zentrum getrenntes Cluster arbeitet unter seinem Warden weiter; ein verlorener vorderer Core lässt die Cluster autonom, bis er aus dem zentralen Core wiederaufgebaut ist. Der Entwurfstest: Kein Ausfall der Steuerungsebene darf jemals die Datenebene degradieren — Helios zu verlieren heißt, die Verwaltbarkeit zu verlieren, niemals die Mission.

13. Reifegrad und weiterer Weg

Die Lieferung richtet sich an Northwinds Phasen aus: zuerst Agent, Warden und vorderer Core mit Enrollment, Abgleich und Telemetrie; dann vollständige Wellenorchestrierung, Out-of-Band-Wiederherstellung und Ereignismaschine; schließlich der föderierte zentrale Core, die Mission-Control-Integration und die Peer-gestützte Verteilung über das Mesh. Offene Fragen sind als Verpflichtungen für den Detailentwurf festgehalten.

Schlussfolgerung

Tangram machte die Enkidu-Gewebe lesbar. Origi machte sie verteidigbar. Helios Grid macht sie betreibbar — und im Flottenmaßstab ist Betreibbarkeit keine Annehmlichkeit, sondern eine Vorbedingung: Tausend ungesteuerte, ungepatchte, unbeobachtete Edge-Knoten sind keine Fähigkeit, sondern ein Passivposten mit Antennen.

Helios' Beitrag ist eine Steuerungsebene, die diszipliniert genug ist, ihr die ganze Flotte anzuvertrauen: Zustand deklariert und gezogen statt geschoben und erhofft; Artefakte signiert, gestaffelt, geprüft und umkehrbar; Sichtbarkeit aus Signal statt aus Daten gebaut, damit die Flotte gesehen werden kann, ohne zu lecken; Behebung nur innerhalb signierter Grenzen automatisiert, mit einkonstruierter menschlicher Naht; und die gesamte Ebene selbst unter Origis Urteil und in Tangrams Struktur stehend, sodass der Betreiber der Flotte an denselben Maßstab gehalten wird wie die Flotte.

Zentral deklarieren. Lokal abgleichen. Nach Ausnahme berichten. Und die Steuerungsebene niemals die Daten berühren lassen.