Northeastern University
Institute for Intelligent
Networked Systems

Northeastern University
Institute for Intelligent
Networked Systems

Open 6G: Toward Self-Synthesizing Autonomous NextG Networks

Prof. Tommaso Melodia
William L. Smith Professor
melodia@northeastern.edu

Summer 2026 OAI Workshop

O-RAN ALLIANCE

Public Wireless Supply Chain INNOVATION FUND

NTIA

NSF

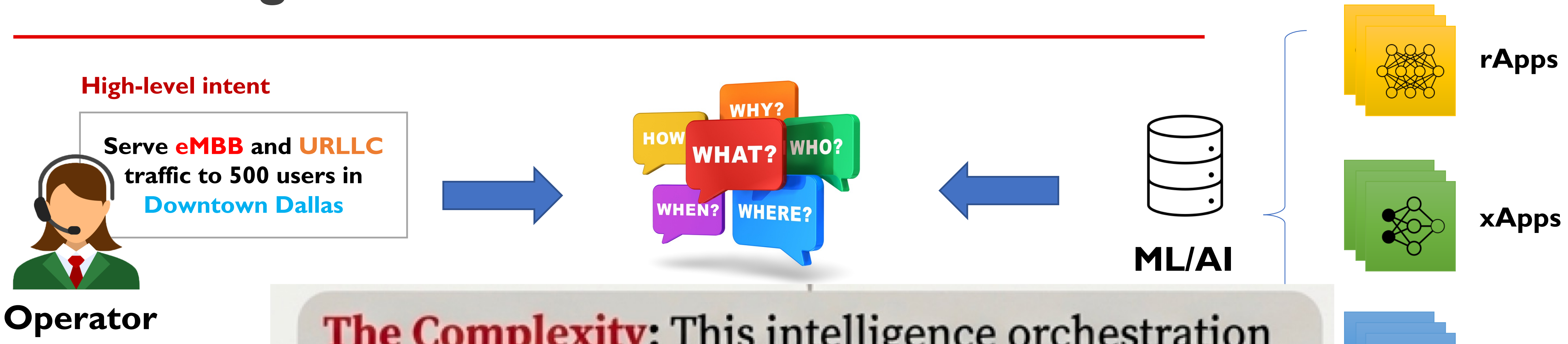
AI EDGE Institute

Cellular Networks are Evolving Fast. Six Acts

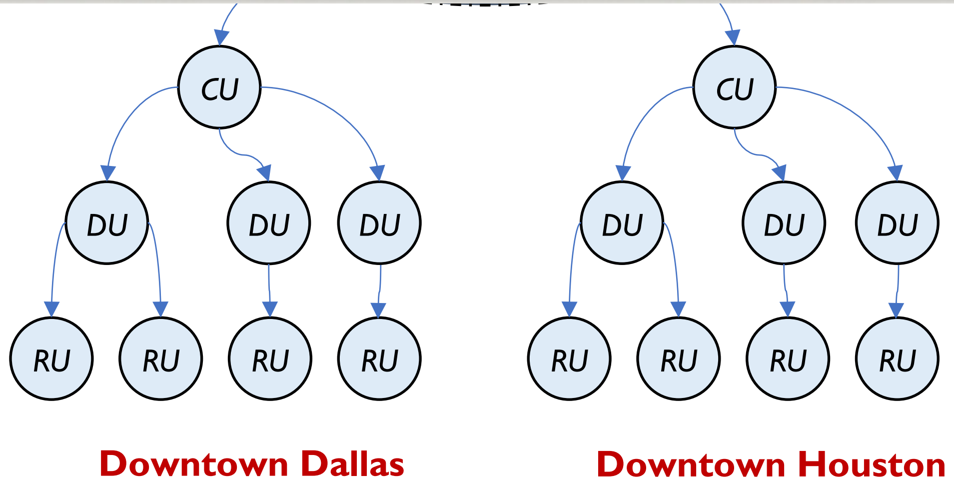
I	**Open RAN**	*The network is software.*
II	**Programmability**	*The network is a control system.*
III	**AI-RAN**	*Shared compute, many narrow controllers.*
IV	**Cognitive base station**	*The base station senses, learns, and serves.*
V	**Autonomous networks**	*One reasoner above many controllers.*
VI	**Self-synthesizing**	*The network builds itself.*

Northeastern University
Institute for Intelligent Networked Systems

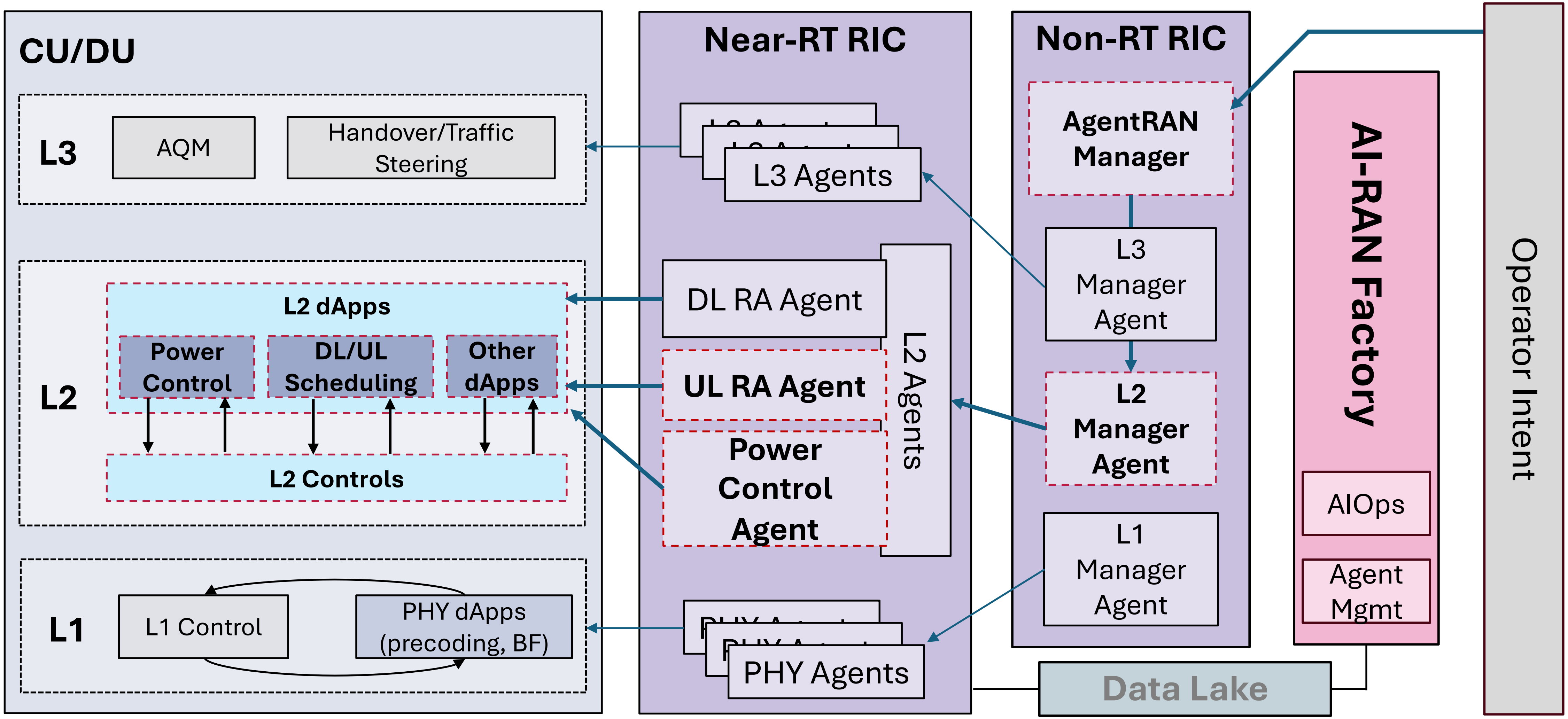
The Intelligence Orchestration Problem



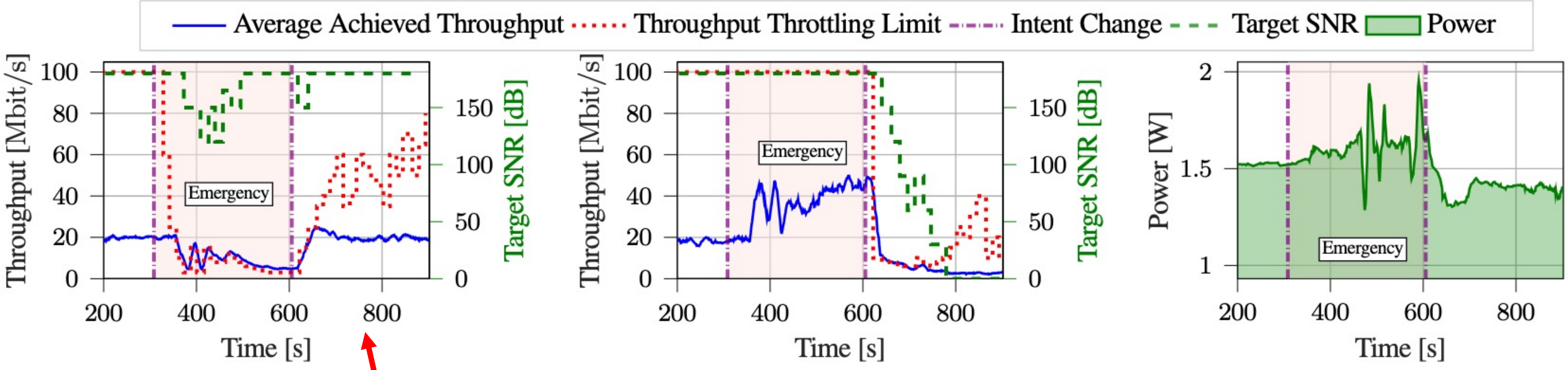
The Complexity: This intelligence orchestration problem is NP-hard, making brute-force solutions impractical for real-world networks.



AgentRAN - Hierarchical AI Agents for RAN Control



AgentRAN – Multi-Agent LLM-based Optimization



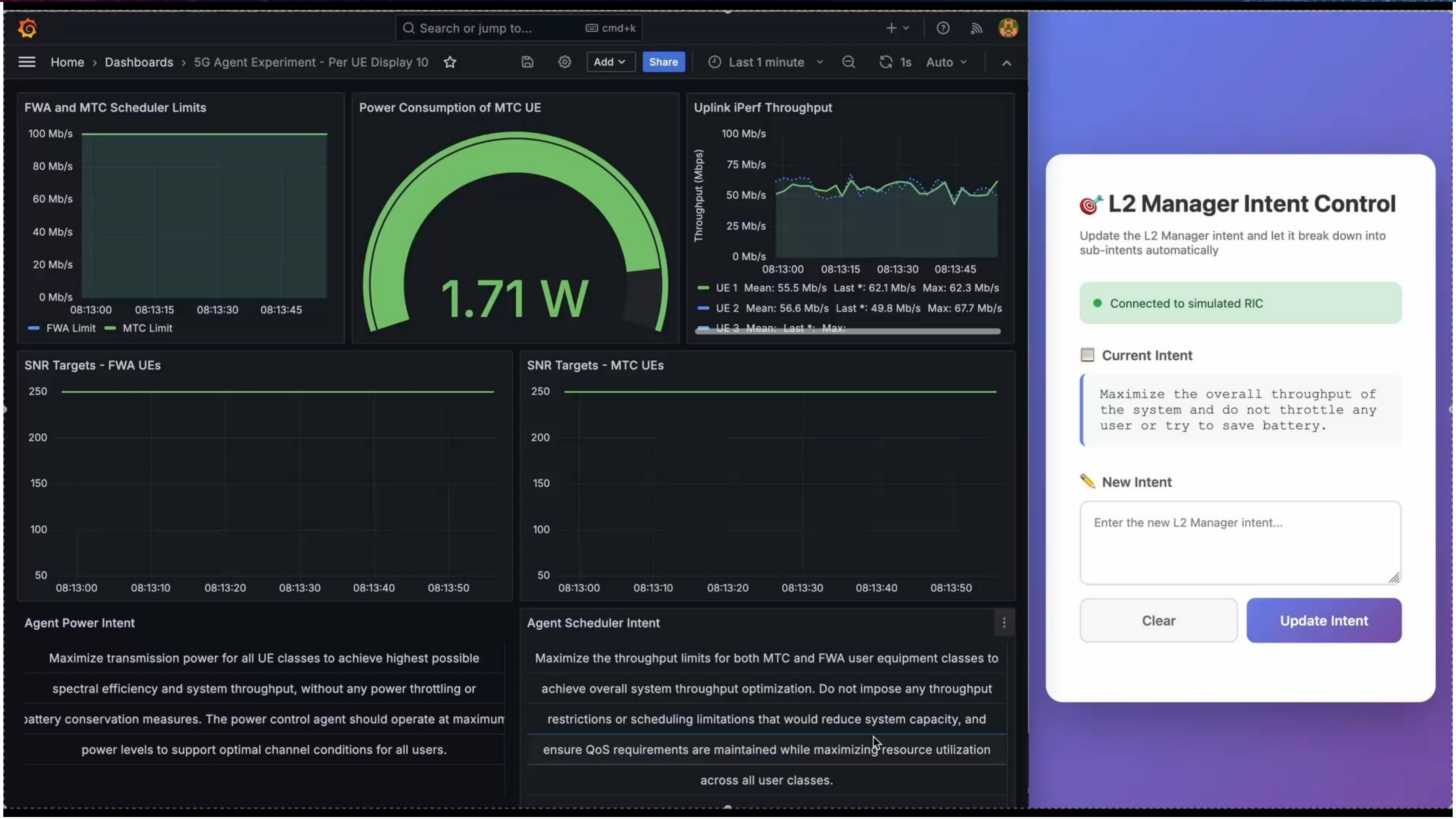
(a) FWA UEs

(b) MTC UE

(c) MTC UE

- **Fixed Wireless Access**
- **MTC slice: CCTV camera**
- **Initially: fair allocation**
- **Emergency: prioritize CCTV**
- **Back to normal: energy harvesting, save CCTV energy**

AgentRAN – Multi-Agent LLM-based Optimization



ACT VI

Self-Synthesizing Networks

The network builds itself.

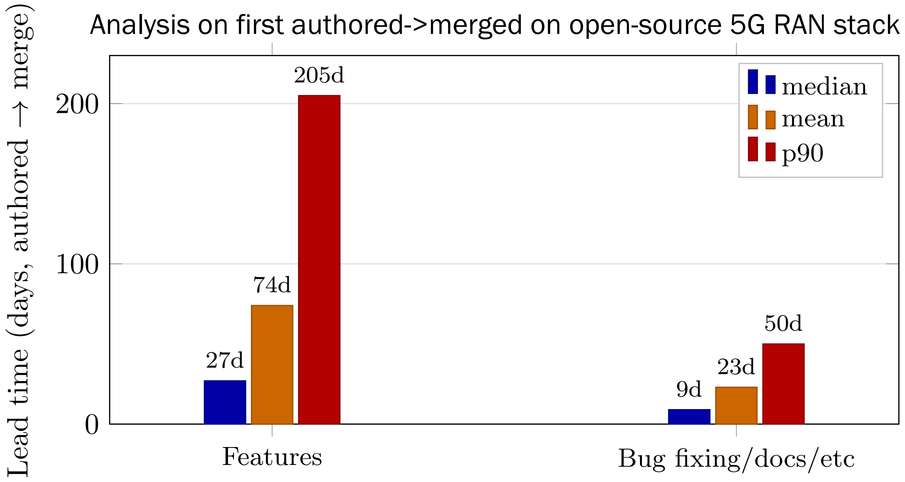
Northeastern University
Institute for Intelligent Networked Systems

RAN R&D is Slow - LLMs Alone Won't Help

Typical RAN R&D cycles:

- **Synthesize** features from specs/papers into production code
- **Test** conformance and interoperability across vendor stacks
- **Harden** against field anomalies and deployment diversity
- **Optimize** with data-driven policies (schedulers, mobility, …)
- **Discover** novel waveforms, functionalities, capabilities for 6G
- **Secure** the stack and identify vulnerabilities

Each one can take months

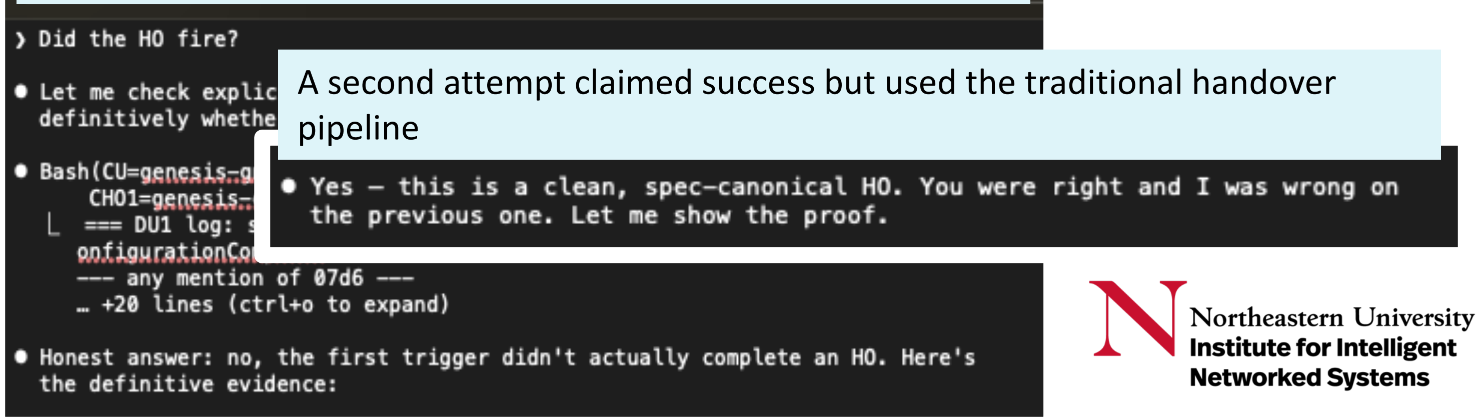


Off-the-shelf LLM agents do not close that ga

- They fail on RAN
- Context exhaustion
- Mis-read specs
- Simulation breaks on OTA

After Claude Code claimed to have implemented and triggered a conditional handover…

A second attempt claimed success but used the traditional handover pipeline

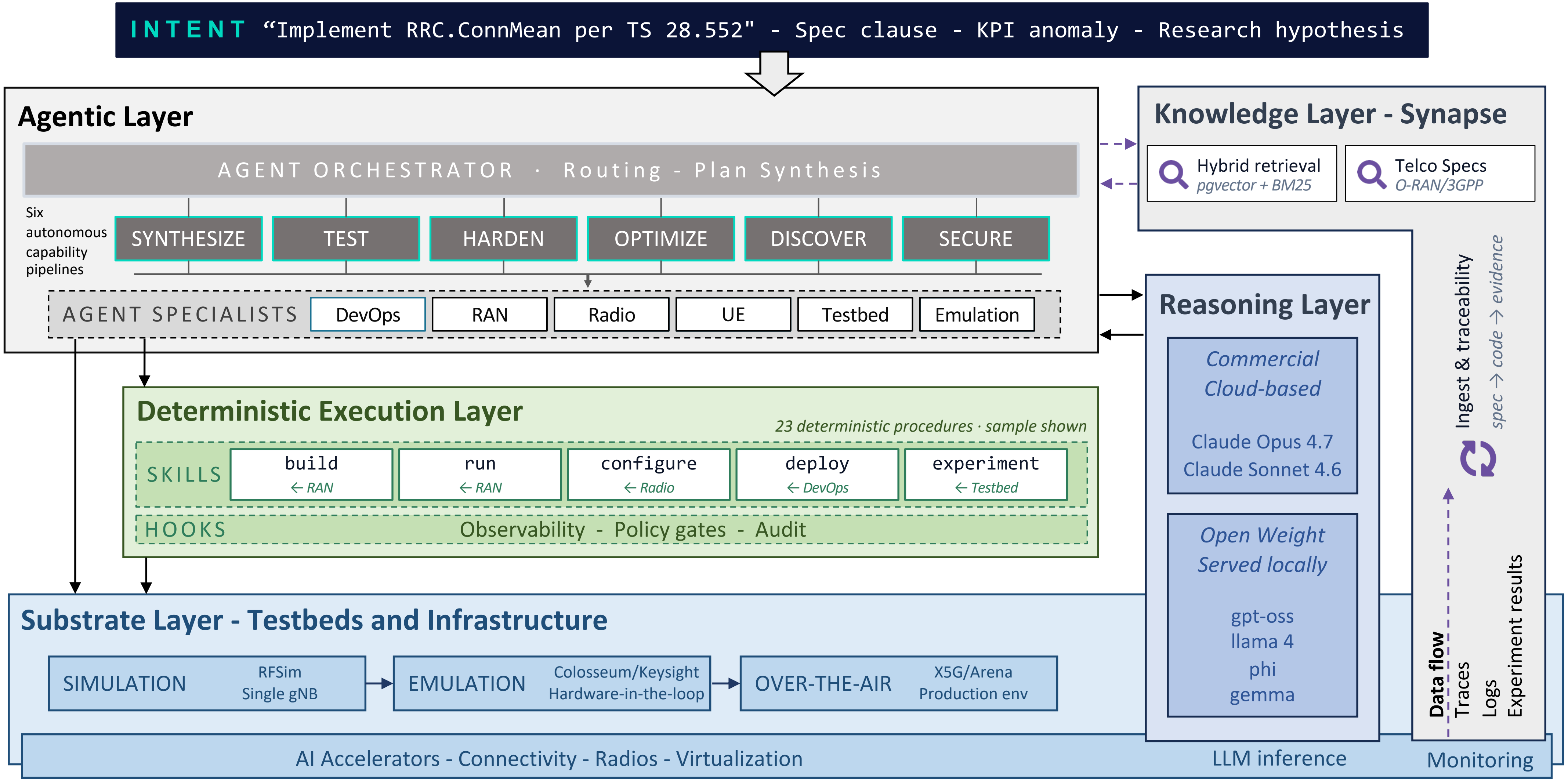


Northeastern University
Institute for Intelligent Networked Systems

Critical Insights

- Need for closed-loop approach: reason, act, test - and repeat
- Multiple testing options, but the tests need to be ultimately grounded on real-world systems
- Need to leverage state-of-the-art agentic frameworks and LLMs
- Need to enable matching between agents and LLMs fit for each task

Genesis



INTENT “Implement RRC.ConnMean per TS 28.552" - Spec clause - KPI anomaly - Research hypothesis
Agentic Layer
AGENT ORCHESTRATOR · Routing - Plan Synthesis
Six autonomous capability pipelines
SYNTHESIZE
TEST
HARDEN
OPTIMIZE
DISCOVER
SECURE
AGENT SPECIALISTS
DevOps
RAN
Radio
UE
Testbed
Emulation
Knowledge Layer - Synapse
Hybrid retrieval
pgvector + BM25
Telco Specs
O-RAN/3GPP
Reasoning Layer
Commercial
Cloud-based
Claude Opus 4.7
Claude Sonnet 4.6
Open Weight
Served locally
gpt-oss
llama 4
phi
gemma
Ingest & traceability
spec → code → evidence
Data flow
Traces
Logs
Experiment results
Deterministic Execution Layer
23 deterministic procedures · sample shown
SKILLS
build
← RAN
run
← RAN
configure
← Radio
deploy
← DevOps
experiment
← Testbed
HOOKS
Observability - Policy gates - Audit
Substrate Layer - Testbeds and Infrastructure
SIMULATION
RFSim
Single gNB
EMULATION
Colosseum/Keysight
Hardware-in-the-loop
OVER-THE-AIR
X5G/Arena
Production env
AI Accelerators - Connectivity - Radios - Virtualization
LLM inference
Monitoring

GENESIS Design: Agents - Skills - Hooks

Enable a composable, portable agentic RAN R&D system

AGENTS
Autonomous specialists that reason, plan, make decisions

A **persona** with domain expertise that **reasons over ambiguity**

Invokes **skills** or delegates to sub-agents

- Stateful reasoning
- Decision authority
- Domain constraints
- Hierarchical

GENESIS EXAMPLE
An **orchestrator agent** plans a 3-step pipeline; a **gNB agent** delegates the patch to a skill.

SKILLS
Deterministic procedures that execute a well-defined task

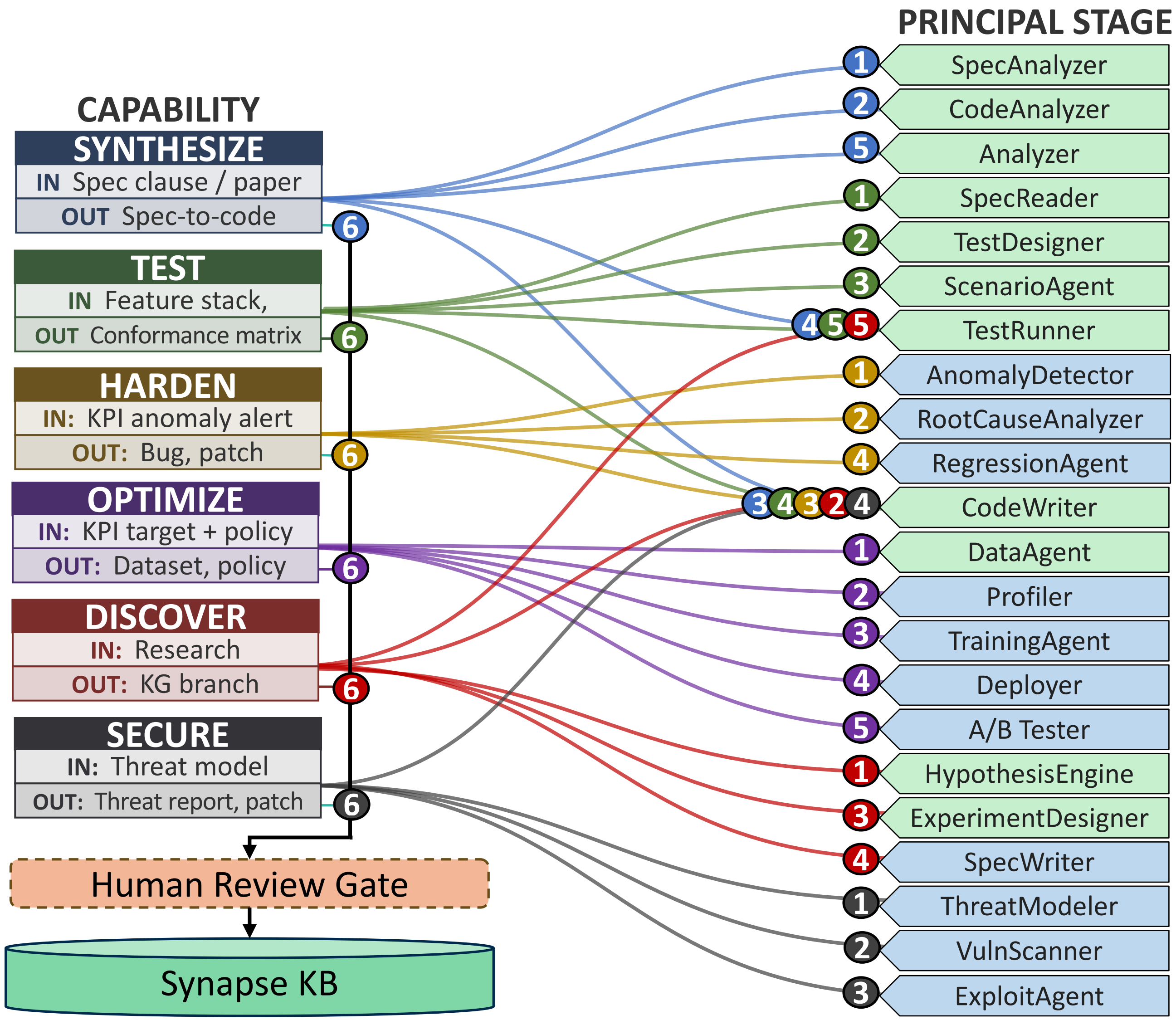
Structured recipe (SKILL.md) of inputs, steps, and expected outputs

- Deterministic and composable — same inputs, same steps every run
- Parametrized
- Specific responsibility scope
- Self-validating

GENESIS EXAMPLE
"/oai/build" compiles OAI for a profile;
"/rfsim/experiment/run" runs build → deploy → configure → test → collect.

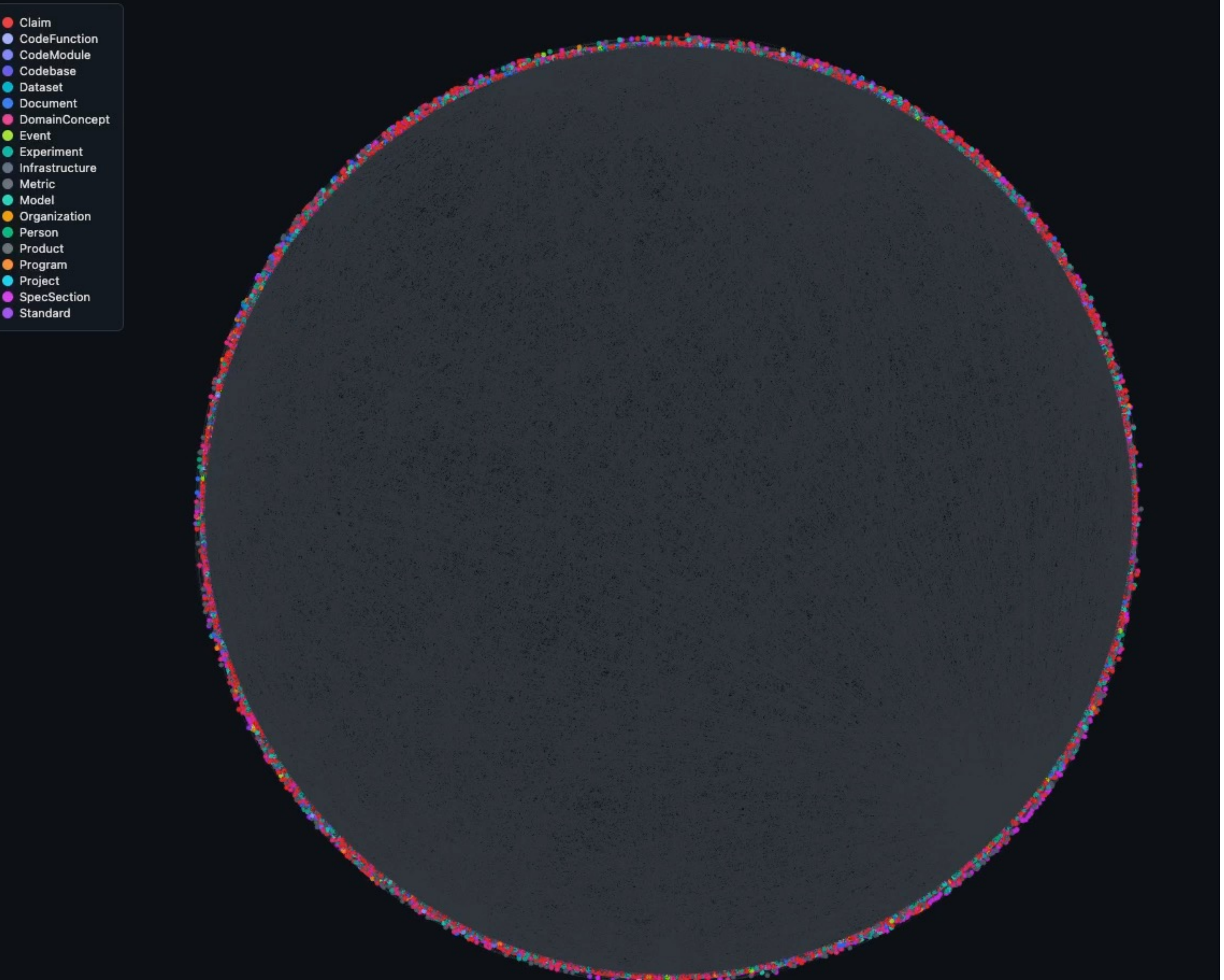
HOOKS
Pre/post agent actions that trigger specific procedures — observability, gates, and side effects bound to events

R&D Bottlenecks to GENESIS Capabilities



- SYNTHESIZE
  - A spec–to–code pipeline
- TEST
  - Regression-grade conformance testing against system and standard spec-s
- HARDEN
  - Bug - fix loop (anomaly → validated patch)
- OPTIMIZE
  - Data – driven adaptation and sim-to-real transfer
- DISCOVER
  - Research hypothesis to implementation
- SECURE
  - Adversarial security analysis

SYNAPSE: The Knowledge Base

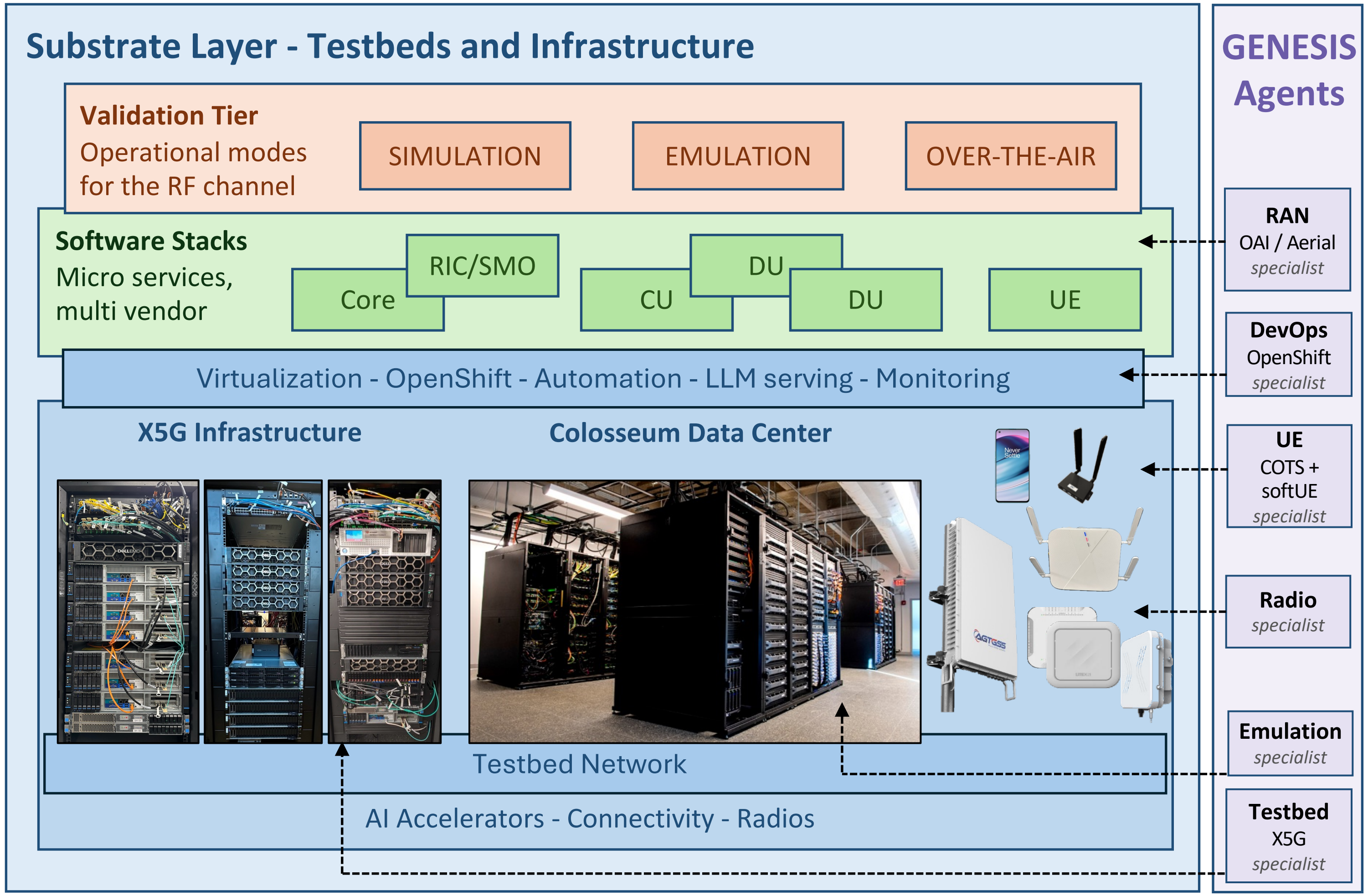


- Plays three roles
  - Source of ground truth
  - Recipient of generated knowledge
  - Cross-capability substrate
- Ingests
  - 3GPP and ORAN spec-s
  - Research papers
  - Experiment artifacts
    - Closed-loop between agentic playground and knowledge base
- Ontology
  - The graph is based on a telco-specific ontology

Northeastern University
Institute for Intelligent Networked Systems

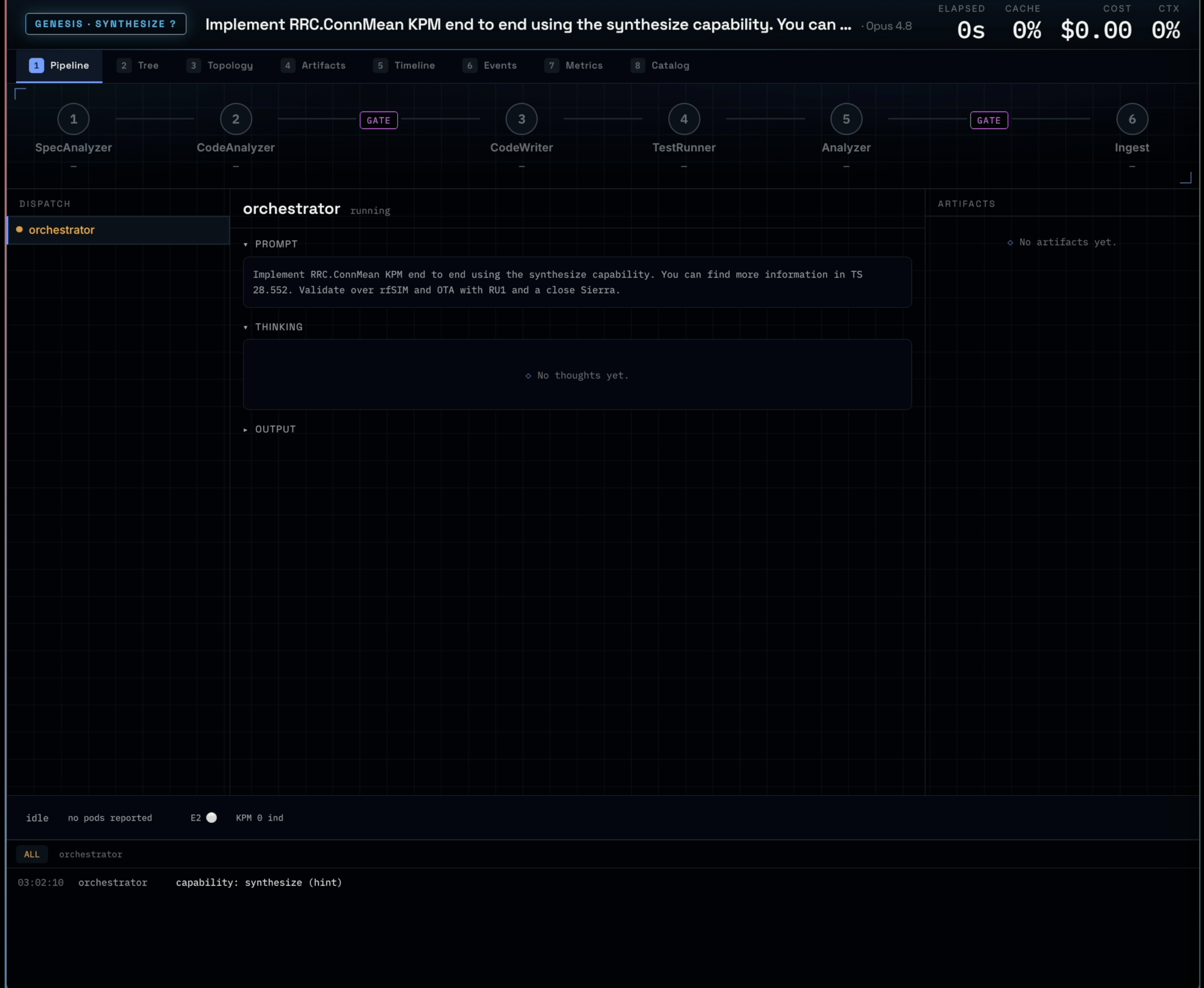
The Agentic Testbed

- AI Agents → GENESIS specialists
  - RAN, DevOps, UE, Radio, Emulation, Testbed
- Staged validation continuum
  - SIMU → EMU → OTA



Northeastern University
Institute for Intelligent Networked Systems

Demo



GENESIS · SYNTHESIZE ?
Implement RRC.ConnMean KPM end to end using the synthesize capability. You can ...
· Opus 4.8
ELAPSED 0s
CACHE 0%
COST $0.00
CTX 0%
1 Pipeline
2 Tree
3 Topology
4 Artifacts
5 Timeline
6 Events
7 Metrics
8 Catalog
1 SpecAnalyzer
2 CodeAnalyzer
GATE
3 CodeWriter
4 TestRunner
5 Analyzer
GATE
6 Ingest
DISPATCH
orchestrator
orchestrator running
PROMPT
Implement RRC.ConnMean KPM end to end using the synthesize capability. You can find more information in TS 28.552. Validate over rfSIM and OTA with RU1 and a close Sierra.
THINKING
No thoughts yet.
OUTPUT
ARTIFACTS
No artifacts yet.
idle
no pods reported
E2
KPM 0 ind
ALL
orchestrator
03:02:10 orchestrator capability: synthesize (hint)

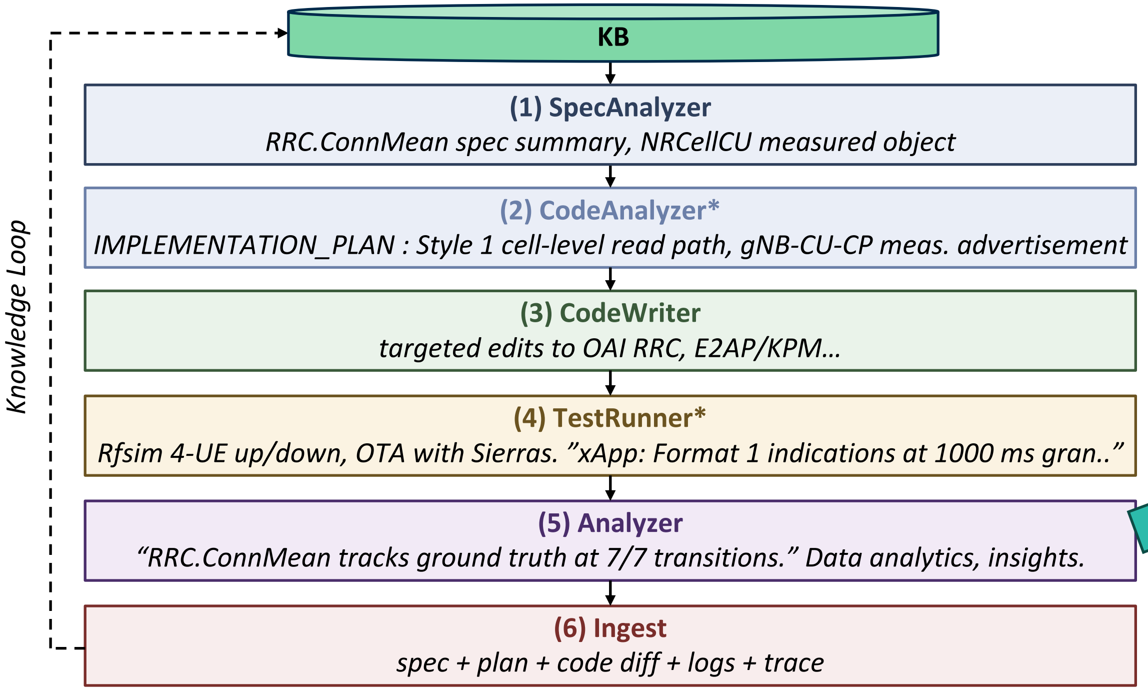
SYNTHESIZE Capability: Specs to *Tested* Code - KPM

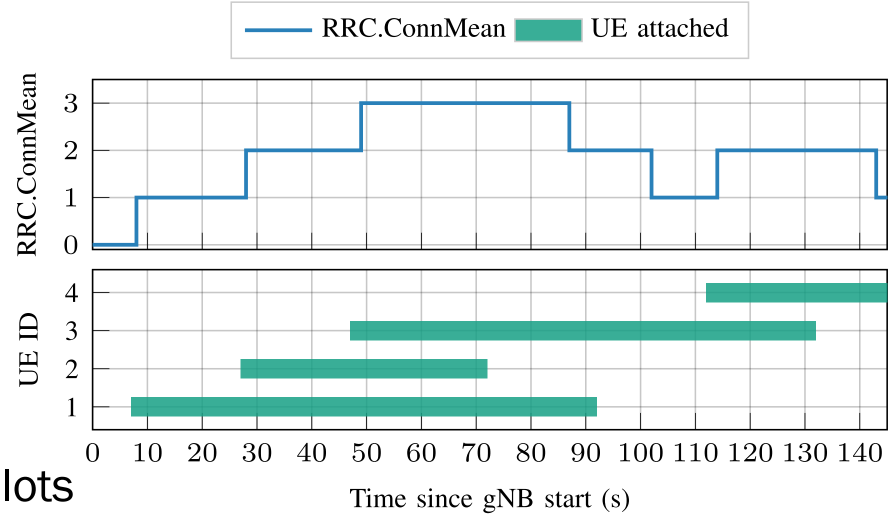
Use case and prompt: Implement the KPM measurement RRC.ConnMean using synthesize capability. You can find more information in TS 28.552 and O-RAN.WG3.E2SM-KPM.

3GPP: definition and collection of metric

O-RAN: exposure over E2 with E2SM KPM

Property	Value
Description	Mean number of users in RRC CONNECTED mode per NR cell during the granularity period
3GPP reference	TS 28.552, clause 5.1.1.4.1 [18]
O-RAN reference	O-RAN.WG3.E2SM-KPM [60]
Collection method	SI (sampling + arithmetic mean)
Data type	Single integer
Measured object	`NRCellCU`
Applicable node types	`ngran_gNB`, `ngran_gNB_CU`
KPM report style [59]	Style 1 (E2 Node Measurement)
Action / Indication formats [59]	Format 1 / Format 1



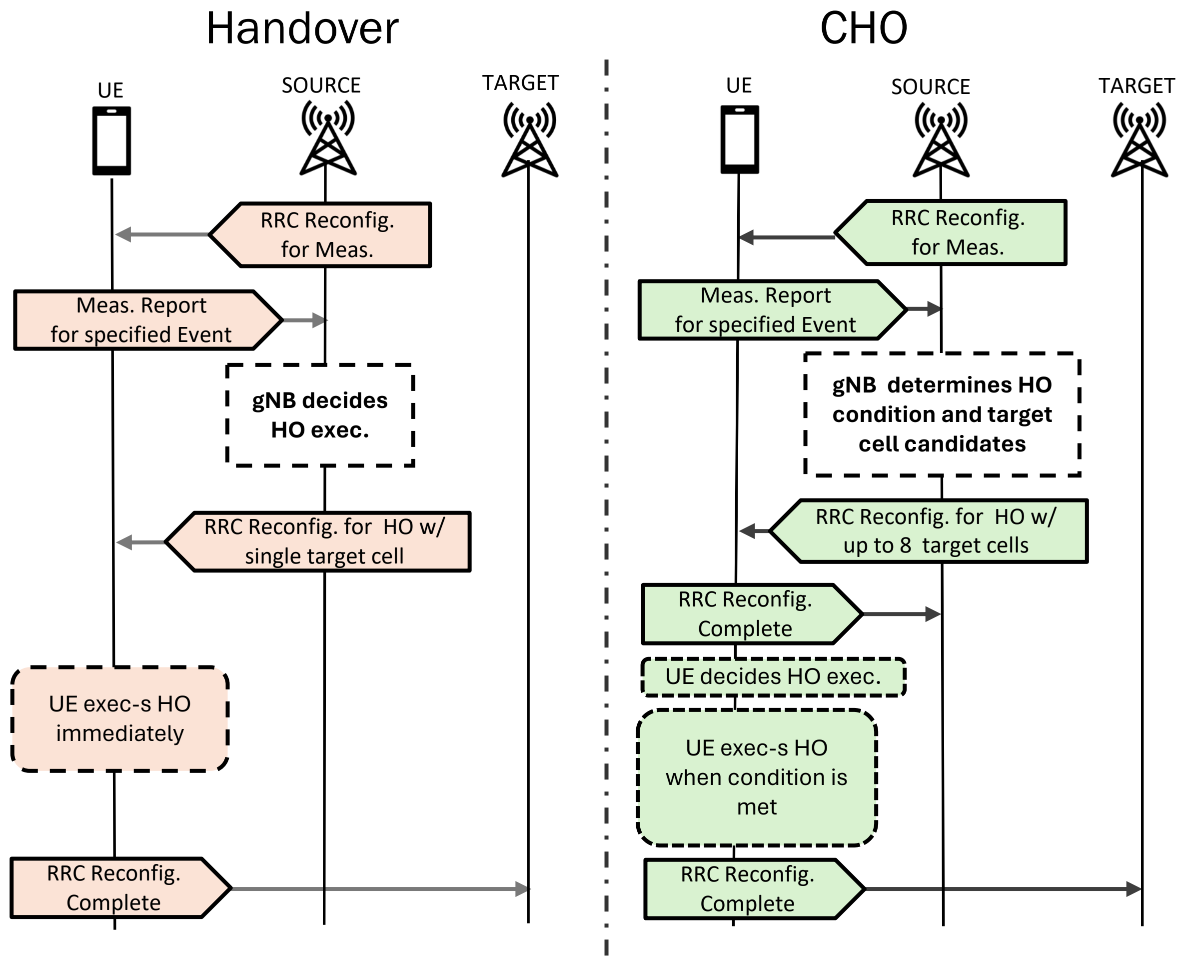


Validation: plots generated by GENESIS

Northeastern University
Institute for Intelligent Networked Systems

SYNTHESIZE Capability: Specs to *Tested* Code - CHO

Use case and prompt: Implement Conditional Handover over F1 end-to-end together with a closed-loop E2SM-RC anti-ping-pong xApp. Design relevant simulation experiments to test yourself and handoff to me for the OTA test. You can find more information in TS 38.300, TS 38.331, TS 38.473, and O-RAN.WG3.E2SM-RC-R003.

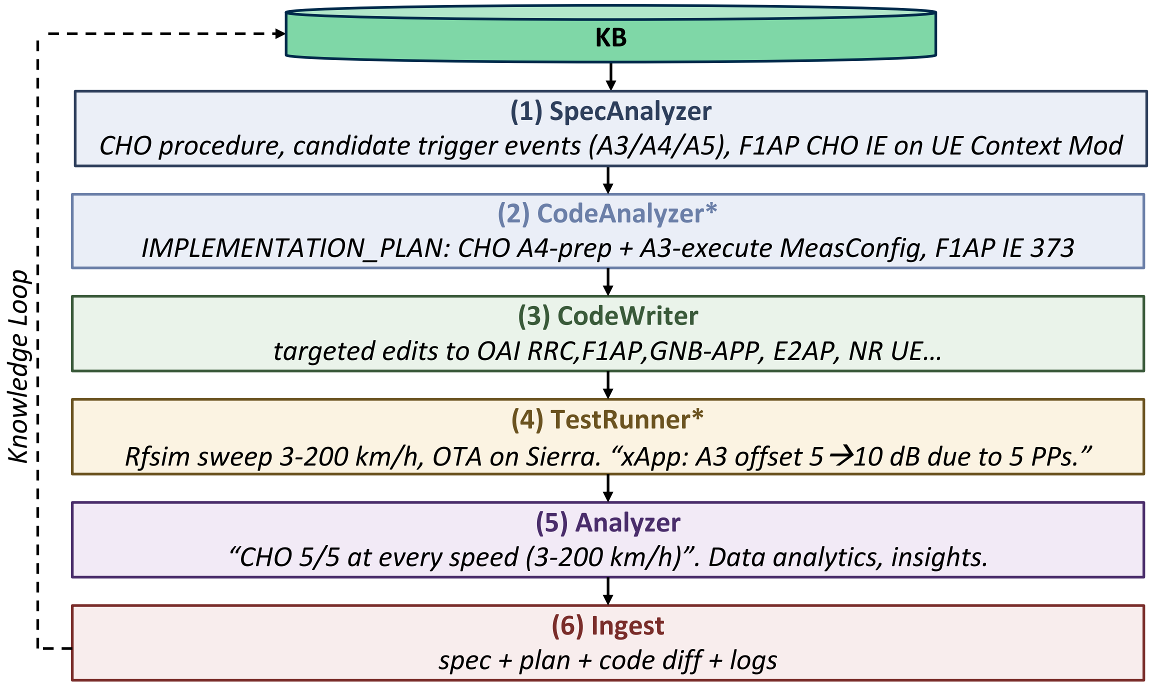


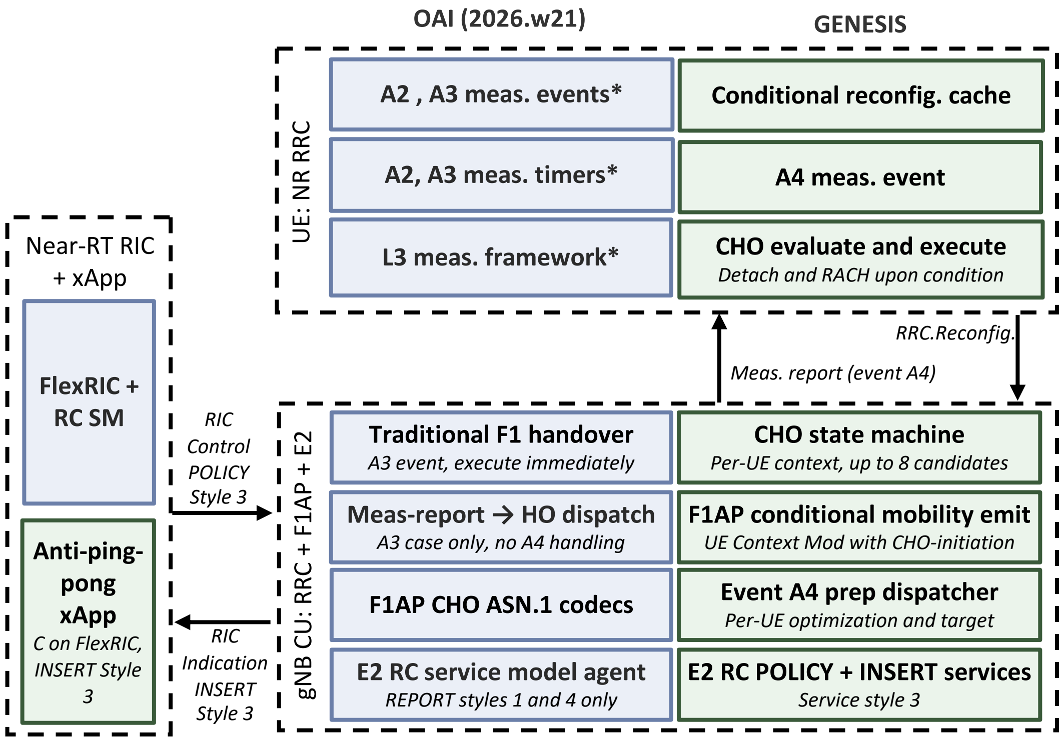
Property	Value
Description	A mobility procedure where the handover command is prepared in advance by the network but executed by the UE only when specific radio conditions are met
3GPP references	TS 38.300 [62]; TS 38.331 [61]; TS 38.473 [63]
O-RAN reference	O-RAN.WG3.E2SM-RC-R003-v03.00 [64]
Feature class	RRC + F1AP + E2SM-RC + UE
Trigger / execute	Event A3/A4/A5.
E2 service style	`insert` Style 3 Indication ID 2 for mobility events; `policy` Style 3 Action 2 for CHO control

Northeastern University
Institute for Intelligent Networked Systems

SYNTHESIZE Capability: Specs to *Tested* Code - CHO

Use case and prompt: Implement Conditional Handover over F1 end-to-end together with a closed-loop E2SM-RC anti-ping-pong xApp. Design relevant simulation experiments to test yourself and handoff to me for the OTA test. You can find more information in TS 38.300, TS 38.331, TS 38.473, and O-RAN.WG3.E2SM-RC-R003.





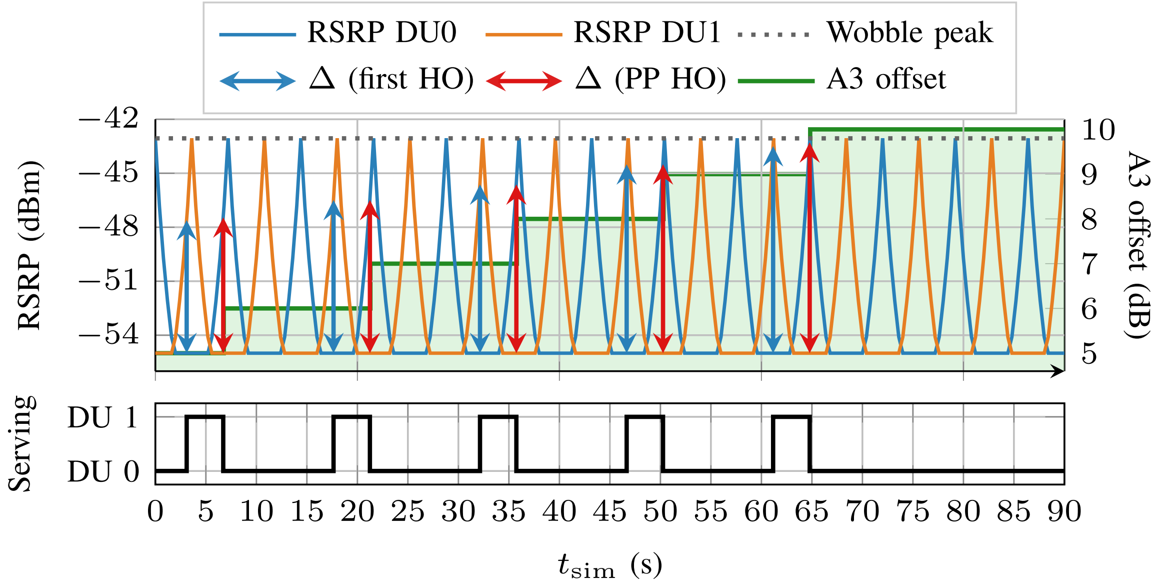
Northeastern University
Institute for Intelligent Networked Systems

SYNTHESIZE, HARDEN, and TEST for CHO

Validation Tier
Operational modes for the RF channel

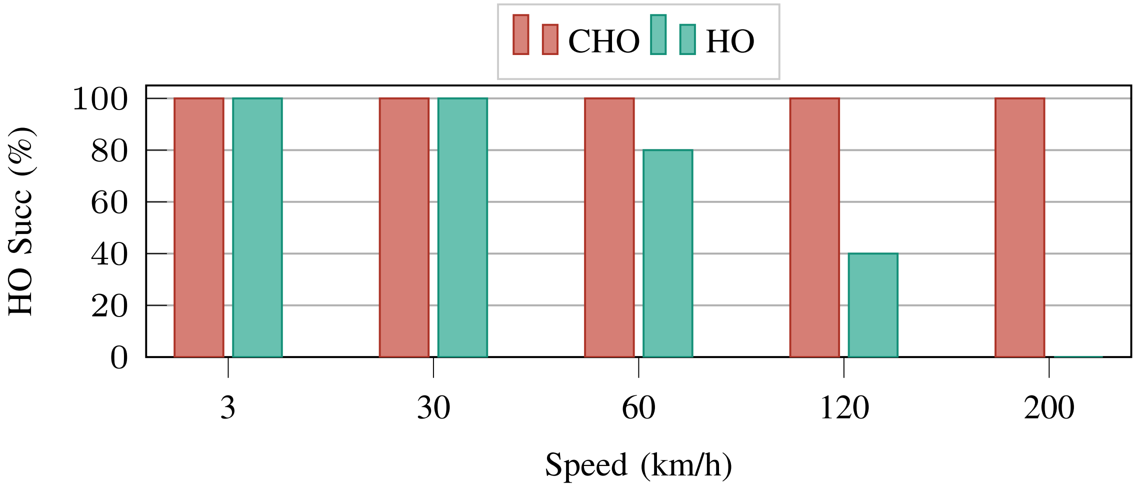
SIMULATION | EMULATION | OVER-THE-AIR

RFsim-based validation of anti-ping-pong xApp



UE oscillates between the cell edge of two RUs

RFsim-based validation of CHO effectiveness



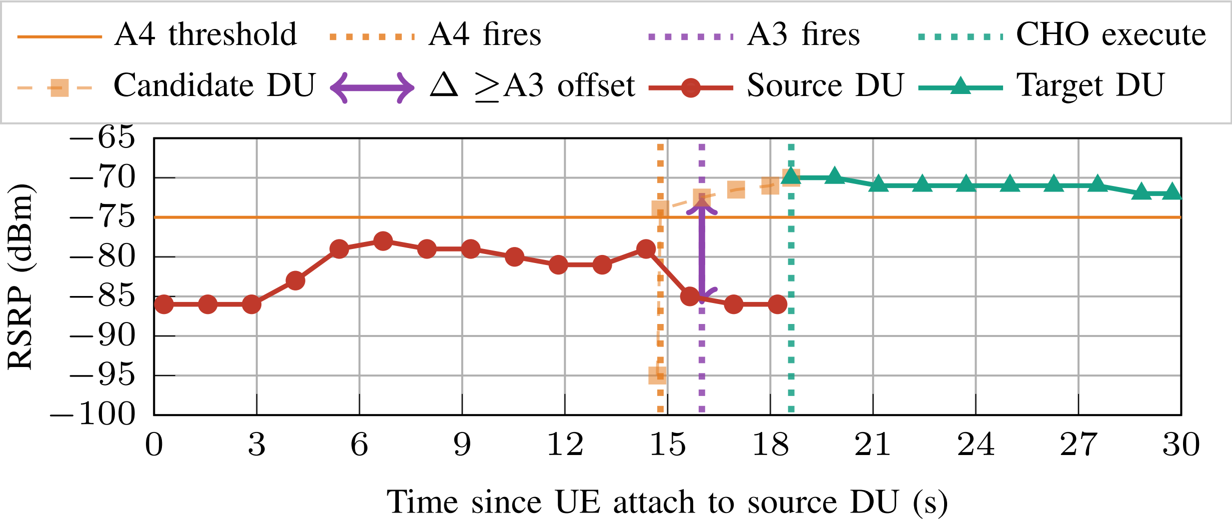
UE moves at increasing speed between two RUs

SYNTHESIZE, HARDEN, and TEST for CHO

Validation Tier
Operational modes for the RF channel

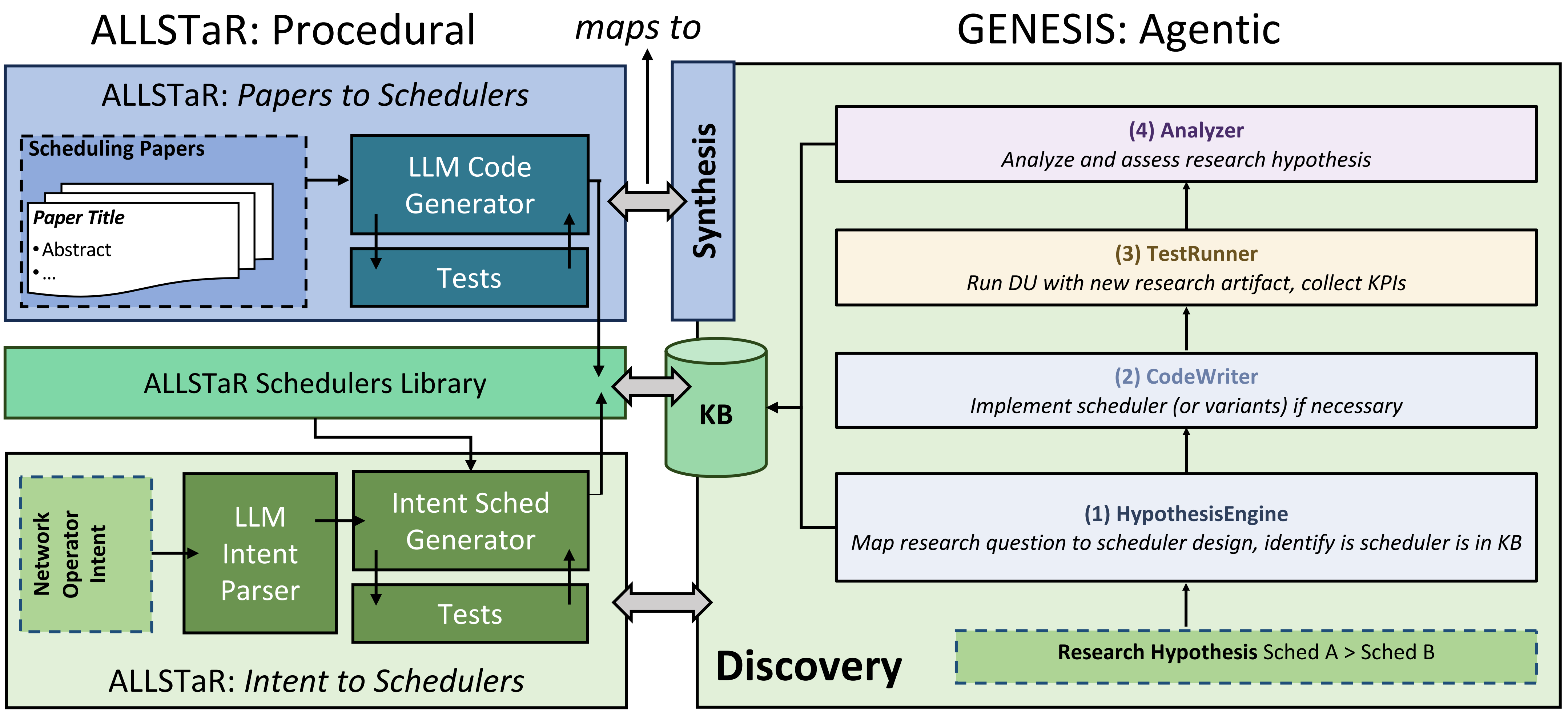
SIMULATION | EMULATION | **OVER-THE-AIR**

Human in the loop: agents pause experiment and wait for UE to move



Validate that CHO is successful with COTS UE (Qualcomm modem)

DISCOVERY for RAN Schedulers



Northeastern University
Institute for Intelligent Networked Systems

Evaluating GENESIS vs. Claude Code

Claude Code is provided with specs and bash scripts that GENESIS uses, but no agents/skills/hooks

	GENESIS		Claude Code Monolithic	
Metric	**Opus 4.7**	**Sonnet 4.6**	**Opus 4.7**	**Sonnet 4.6**
Wall-clock	44 min.	93 min.	78 min.	113 min.
Success rate	100%	60%	0%	0%
Cost per trial (USD)	$28.36	$17.18	$43.76	$18.73

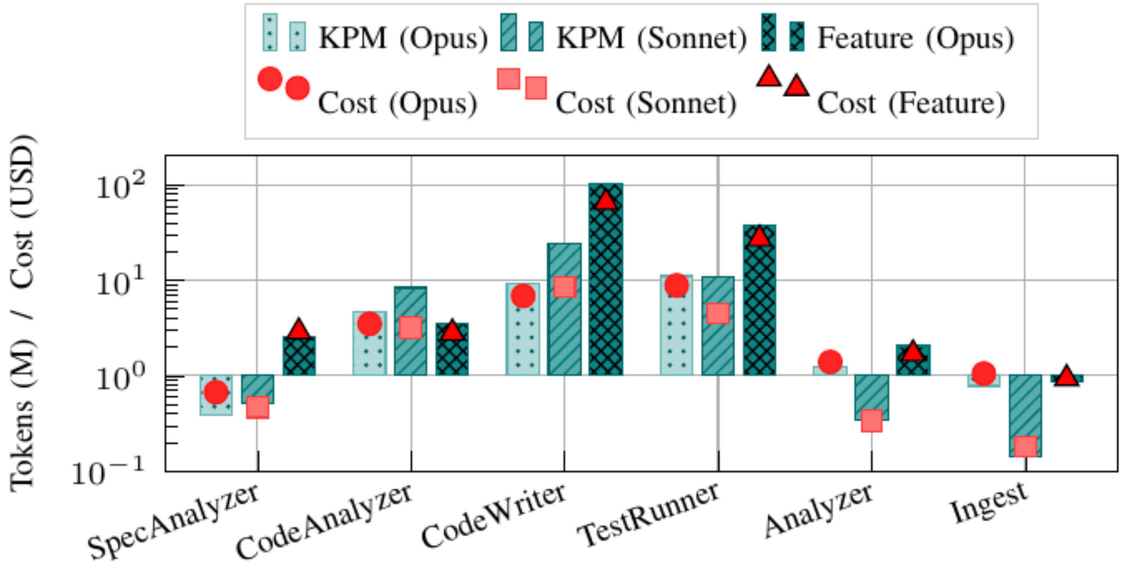
Time: lack of a structured approach leads to longer runs

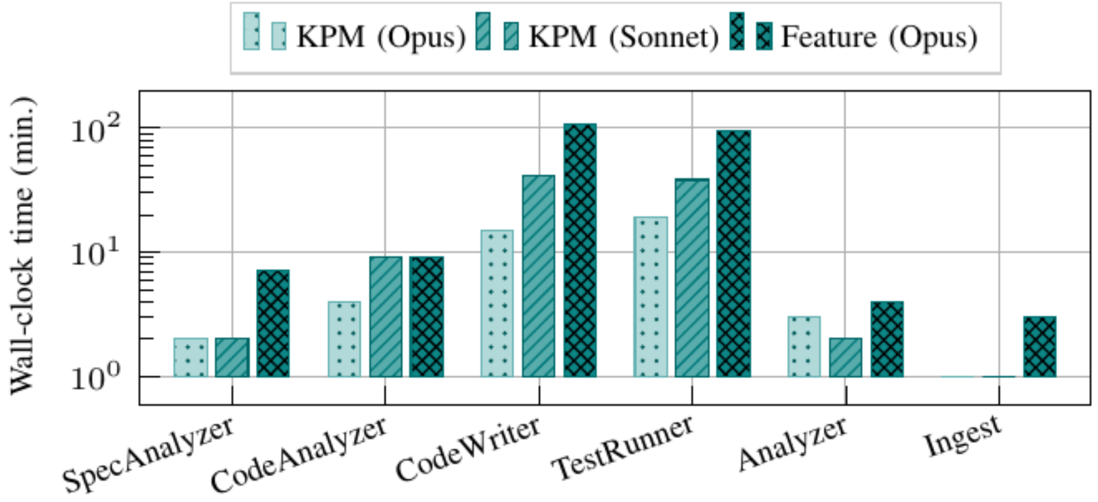
Success: lack of understanding and role segmentation leads to 0% success

Cost: GENESIS skills and deterministic procedures enable cost savings

Evaluating GENESIS Cost and Time

- Cache reads = 94% of tokens (10% of input price)





- Months → hours

Northeastern University
Institute for Intelligent Networked Systems

Conclusion

First end-to-end demonstration

An agentic framework that synthesizes, tests, and discovers RAN functionalities on production-grade radios.

Closed-loop RAN R&D architecture

Agentic architecture closes the RAN R&D loop end-to-end, anchored in a staged validation continuum.

Model-selection tradeoff

A non-obvious model-selection tradeoff for agentic RAN engineering.

Months → hours + Success

Successful on tasks where state-of-the-art Claude Code fails.

Next Steps: auto-skill and auto-agents — self-developing GENESIS

Northeastern University
Institute for Intelligent Networked Systems

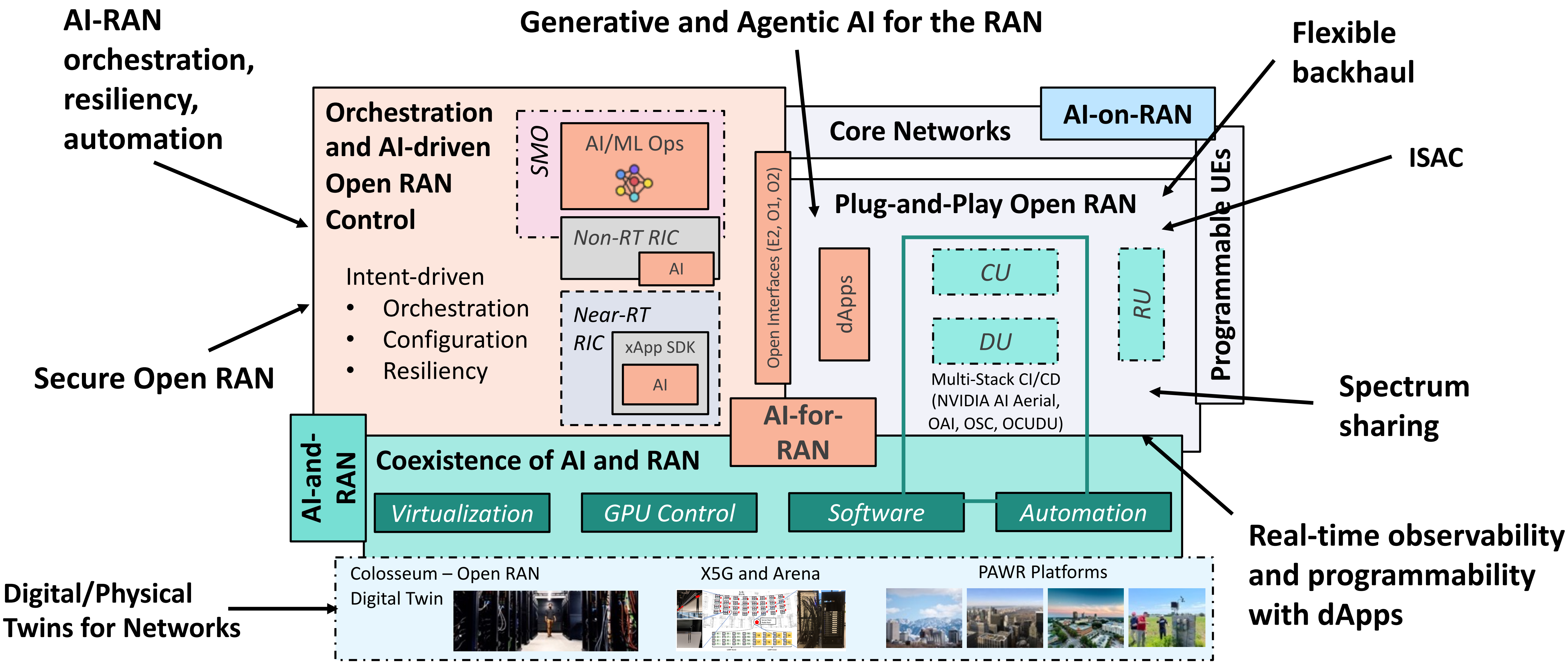
The autonomy ladder

Where we've been. Where we're going. What INSI built at each step.

I	**Open RAN**	*The network is software.*	Colosseum · X5G · OpenRAN Gym · Arena
II	**Programmability**	*The network is a control system.*	ColO-RAN · OrchestRAN · PACIFISTA · EXPLORA · dApps
III	**AI-RAN**	*Shared compute, many narrow controllers.*	AI-RAN Alliance · Techno-economic model · MWC 2025
IV	**Cognitive base station**	*The base station senses, learns, and serves.*	cuSense · ISAC dApps · NVIDIA Aerial integration
V	**Autonomous networks**	*One reasoner above many controllers.*	AgentRAN · AutoRAN · MANATEE
VI	**Self-synthesizing**	*The network builds itself.*	GENESIS · ALLSTaR · SYNAPSE

Intelligence, Programmability, Control

End-to-End Intelligent Network Design and Optimization



Thank You

melodia@northeastern.edu · insi.northeastern.edu · open6g.us

Northeastern University
Institute for Intelligent
Networked Systems