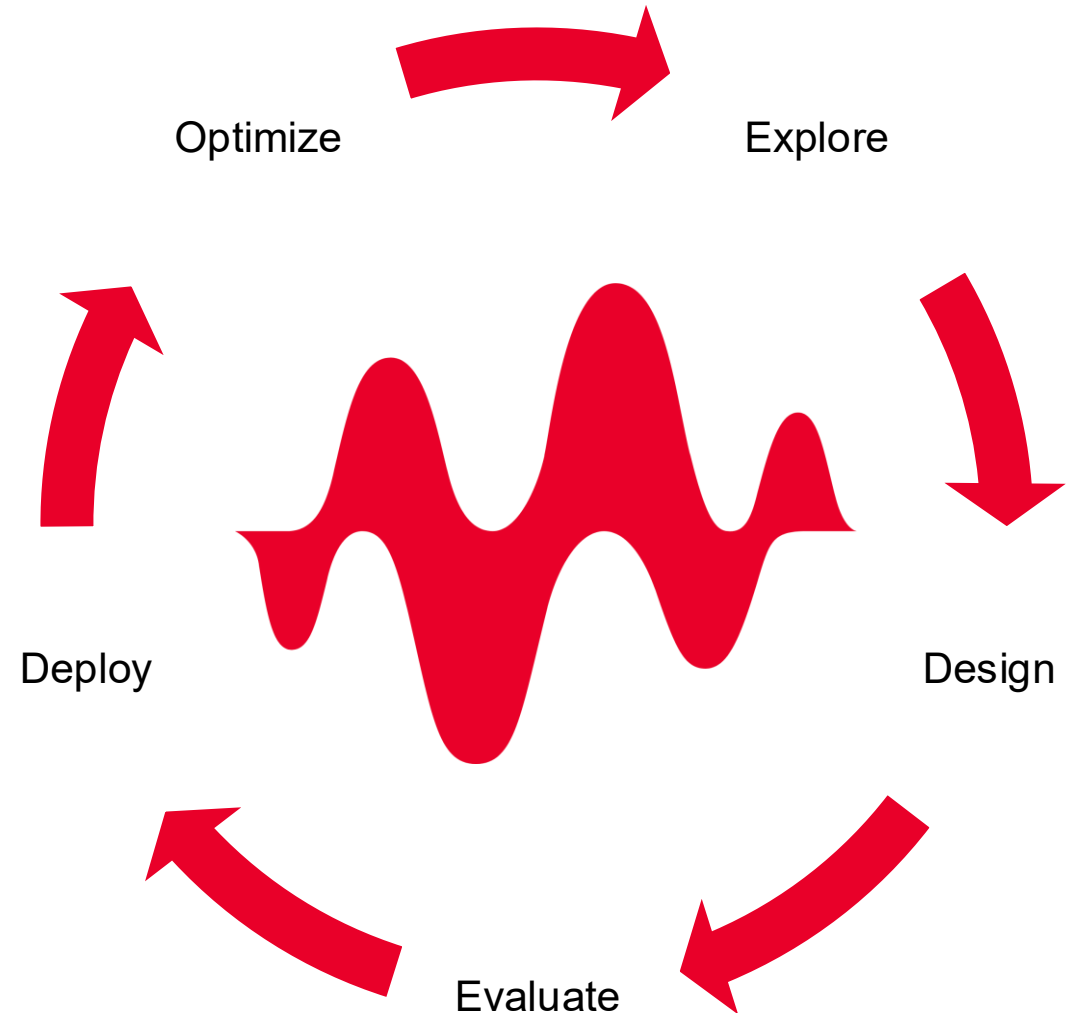


ASTRA-sim-service: A Composable Simulation Interface for AI Infrastructure Co-Design Across Workloads and Fidelity Tiers

Harsh Sikhwal
Keysight Technologies

How We View the AI Design Cycle

At Keysight, we view AI system design as an iterative process rather than a linear sequence of steps. As systems progress from exploration to deployment, insights from simulation, emulation, software, and hardware collectively contribute to continuous learning and optimization.

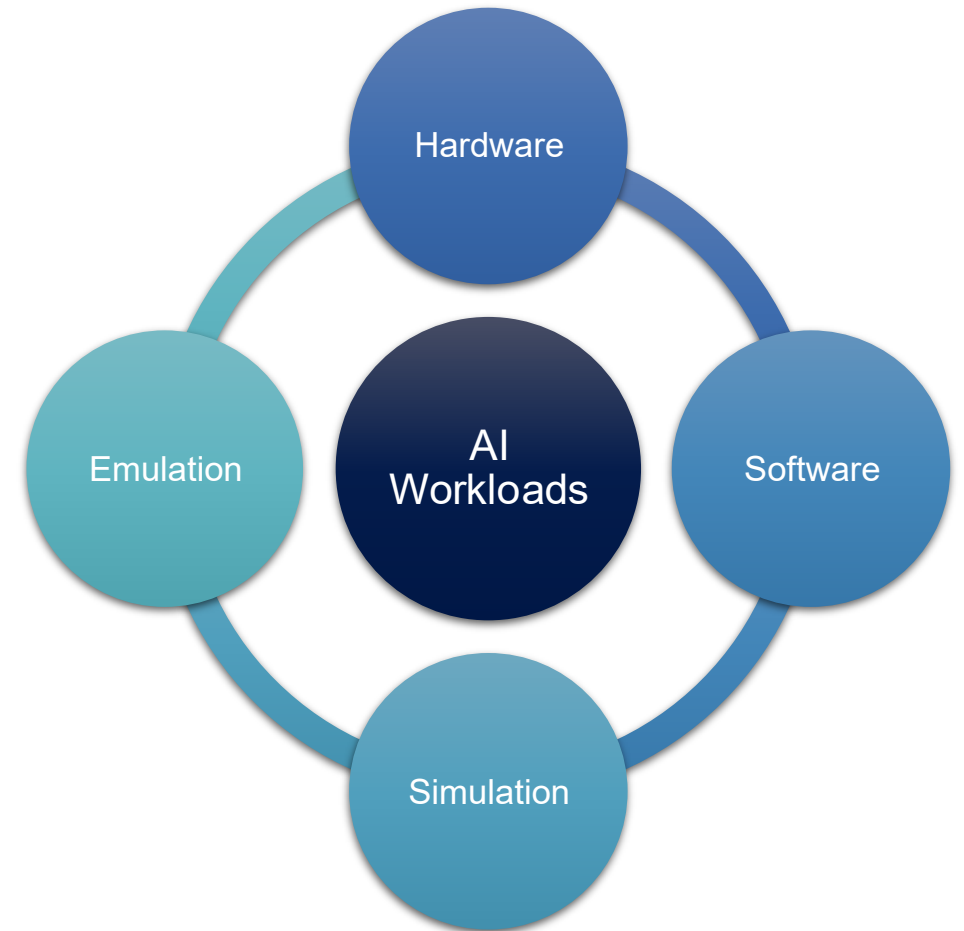


AI system design is an iterative process spanning multiple perspectives

AI Workloads Connect These Perspectives

AI workloads naturally span multiple methodologies throughout their lifecycle

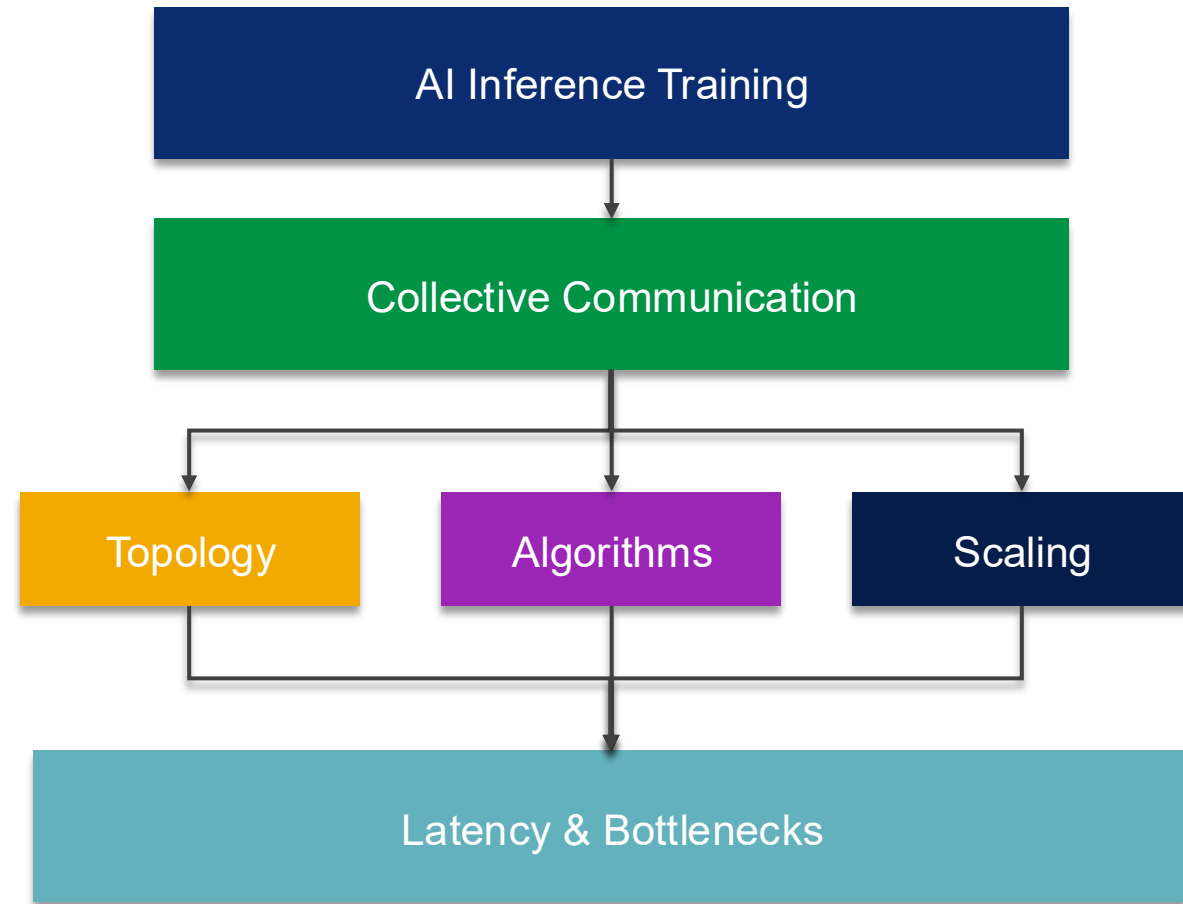
- **Workloads Evolve**
Questions change as systems mature
- **Methodologies Complement One Another**
No single perspective provides the complete picture
- **Insights Flow Across the Cycle**
Results from one stage influence decisions in others.



Collective Communication as a Lens into AI Systems

Many AI system questions ultimately manifest as communication problems.

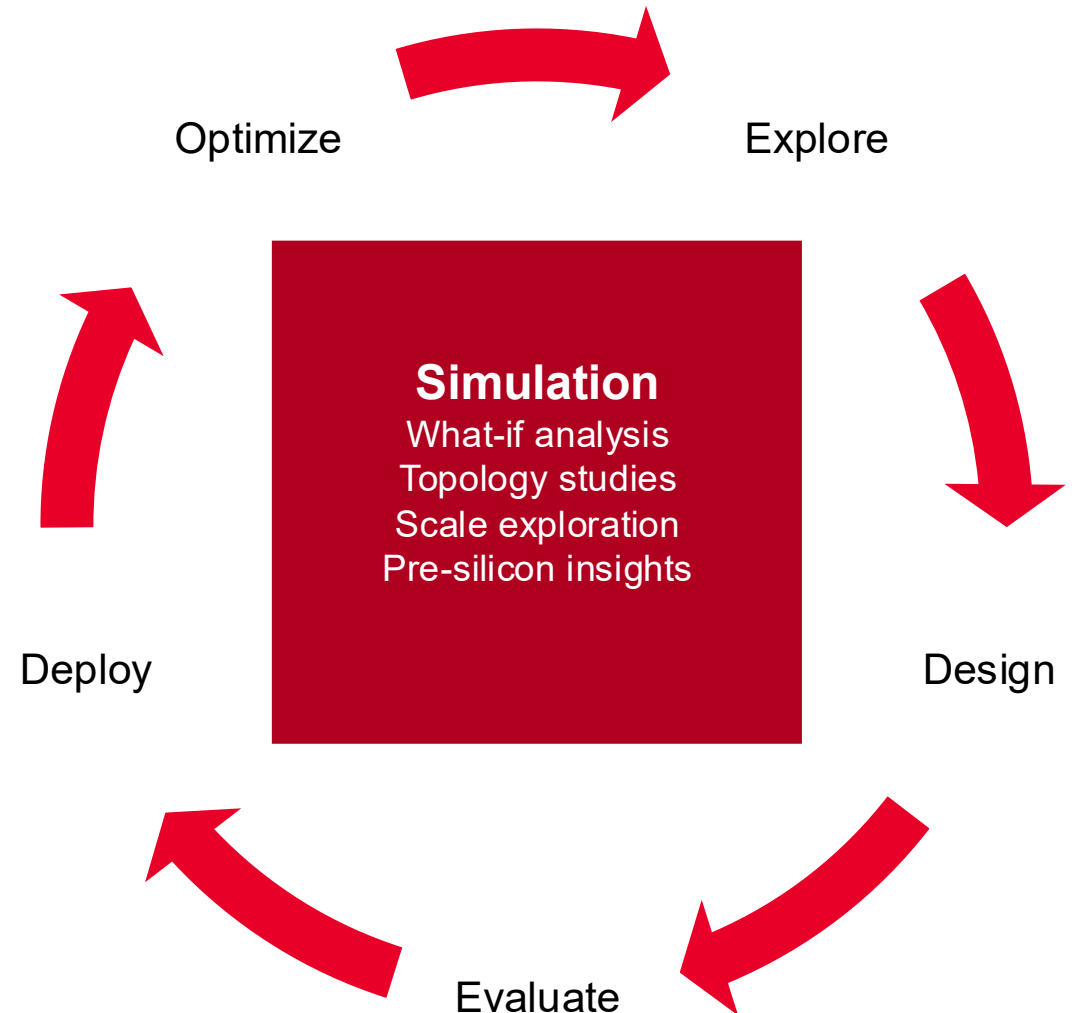
- **Scale and Efficiency**
How well do systems scale?
Where are the communication bottlenecks?
- **Fabric and Topology**
Which network architectures perform best?
How do deployment choices affect performance?
- **Communication–Compute Interaction**
How do computation and communication overlap?
What limits end-to-end efficiency?
- **Cross-System Communication**
Multi-cluster deployments
Distributed training environments



Why Astra-sim Became Valuable

Simulation enables exploration before hardware exists

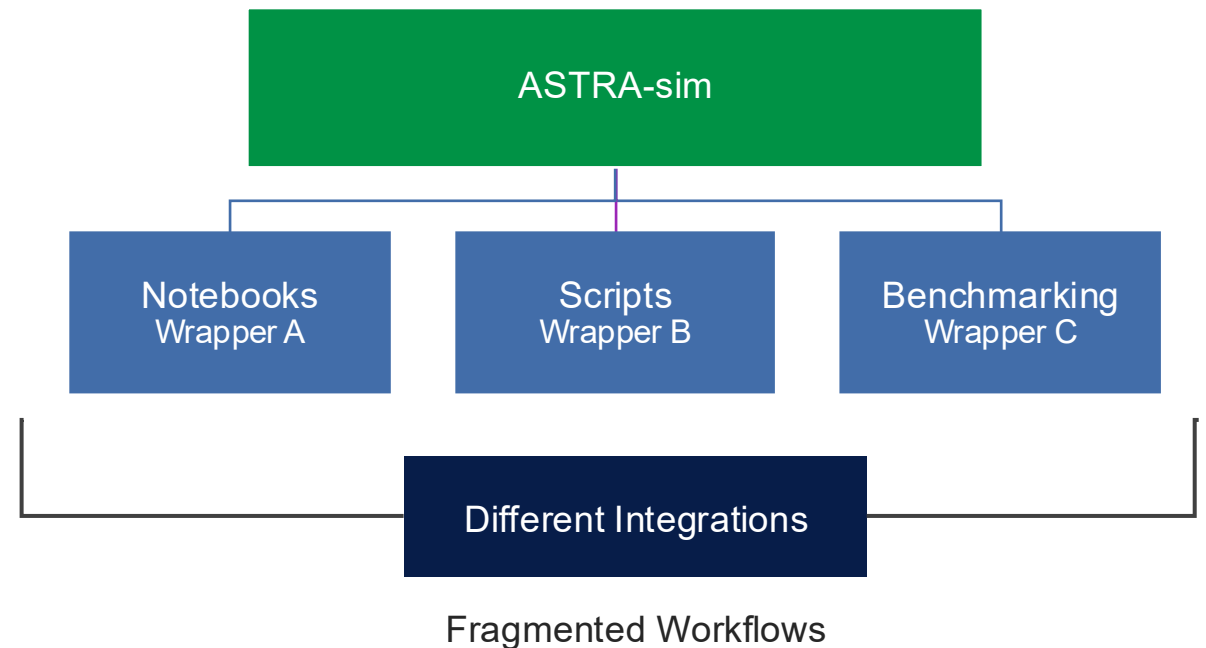
- Explore collective communication before hardware exists
- Compare topology, fabric, and deployment alternatives
- Study scaling behavior of collectives across systems
- Evaluate bandwidth and latency sensitivity
- Explore cross-cluster / DCI collective communication
- Run repeatable what-if experiments



The Integration Challenge

Simulation wasn't the challenge—the integration was

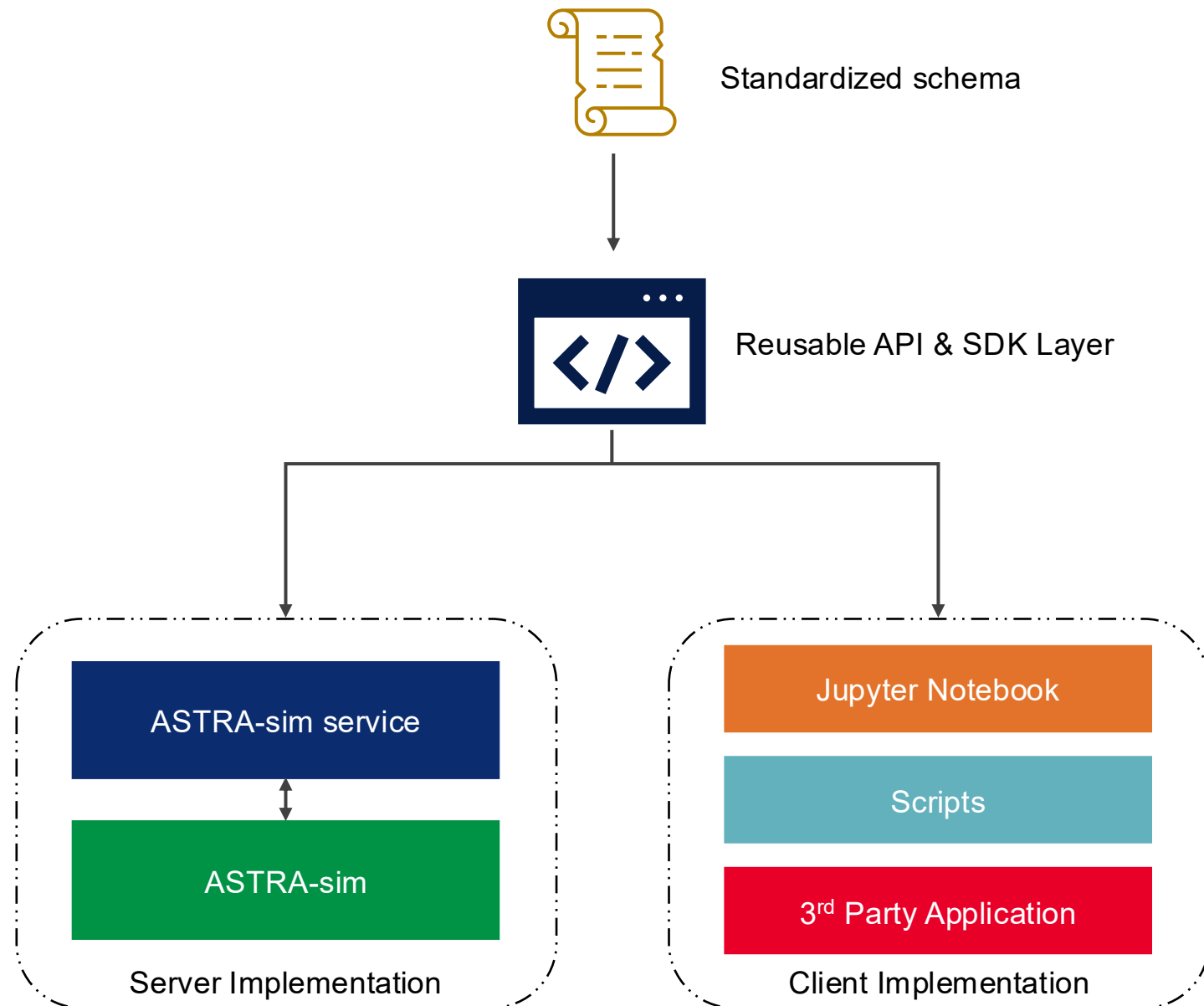
- **Repeated Configuration**
Same setup recreated in notebooks and scripts.
- **Custom Integrations**
Each workflow required its own wrapper.
- **Limited Reusability**
Experiments stayed tied to individual applications.
- **Need for a Common Interface**
Simulation lacked a reusable access layer.



A Common Interface to Simulation

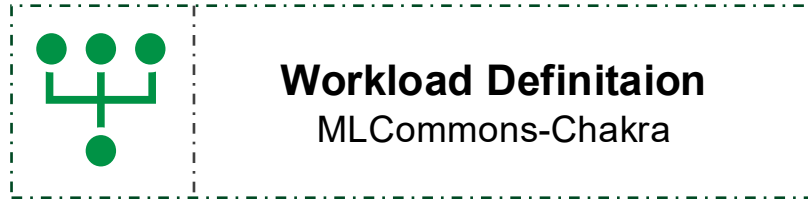
Introducing Astra-sim-service

1. Standardized APIs and schema
2. Reusable API Layer
3. Lightweight service
4. Pluggable into different applications



ASTRA-sim-service

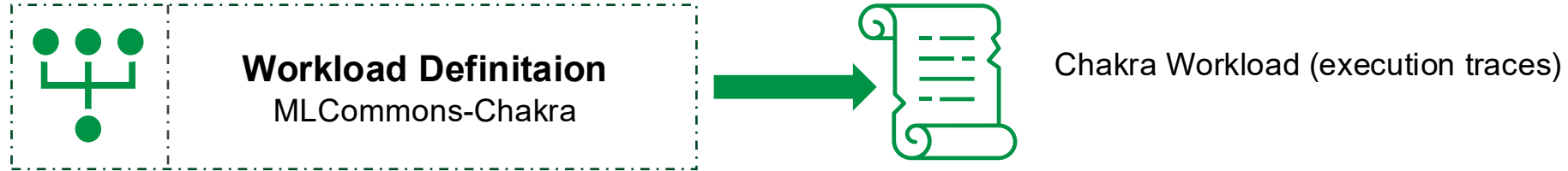
Schemas provide a common representation for these inputs



ASTRA-sim

ASTRA-sim-service

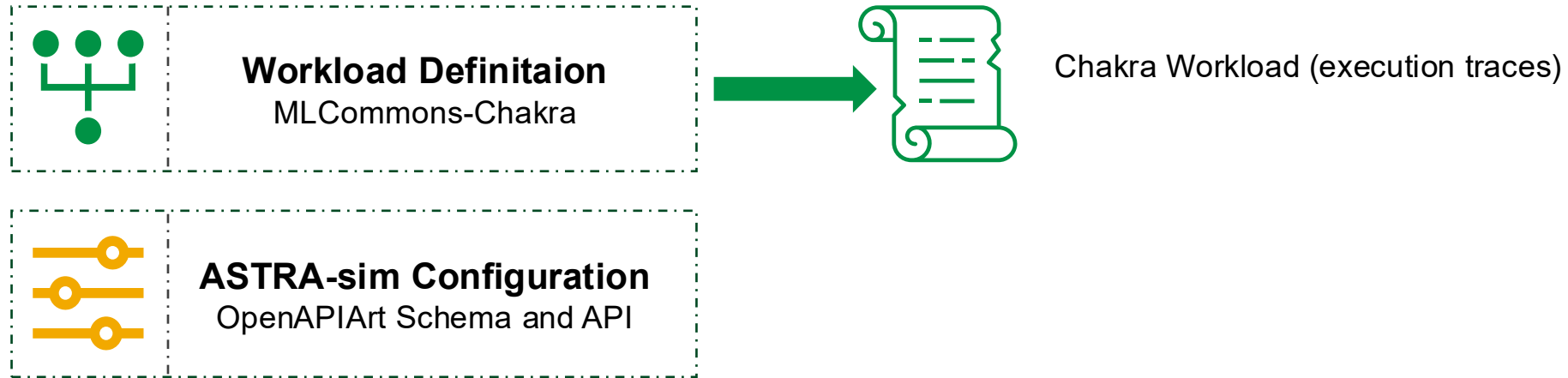
Schemas provide a common representation for these inputs



ASTRA-sim

ASTRA-sim-service

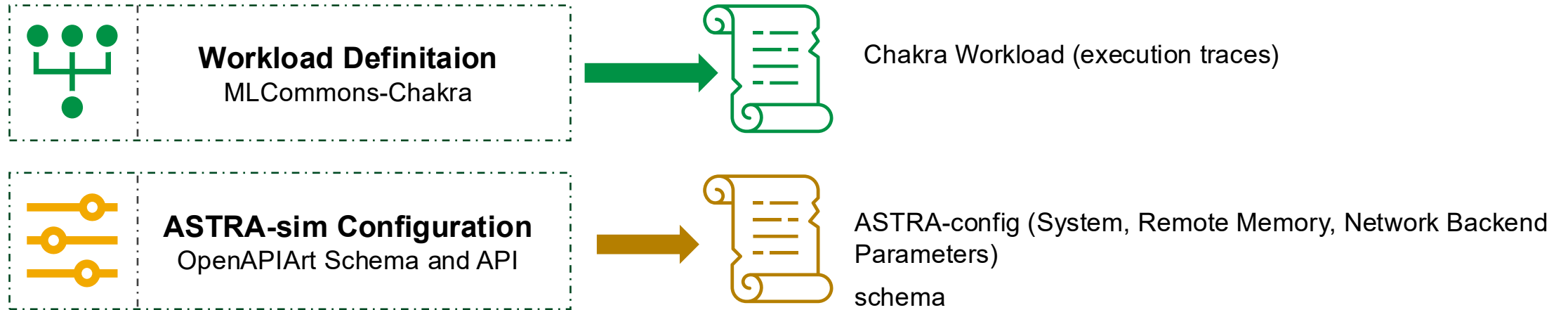
Schemas provide a common representation for these inputs



ASTRA-sim

ASTRA-sim-service

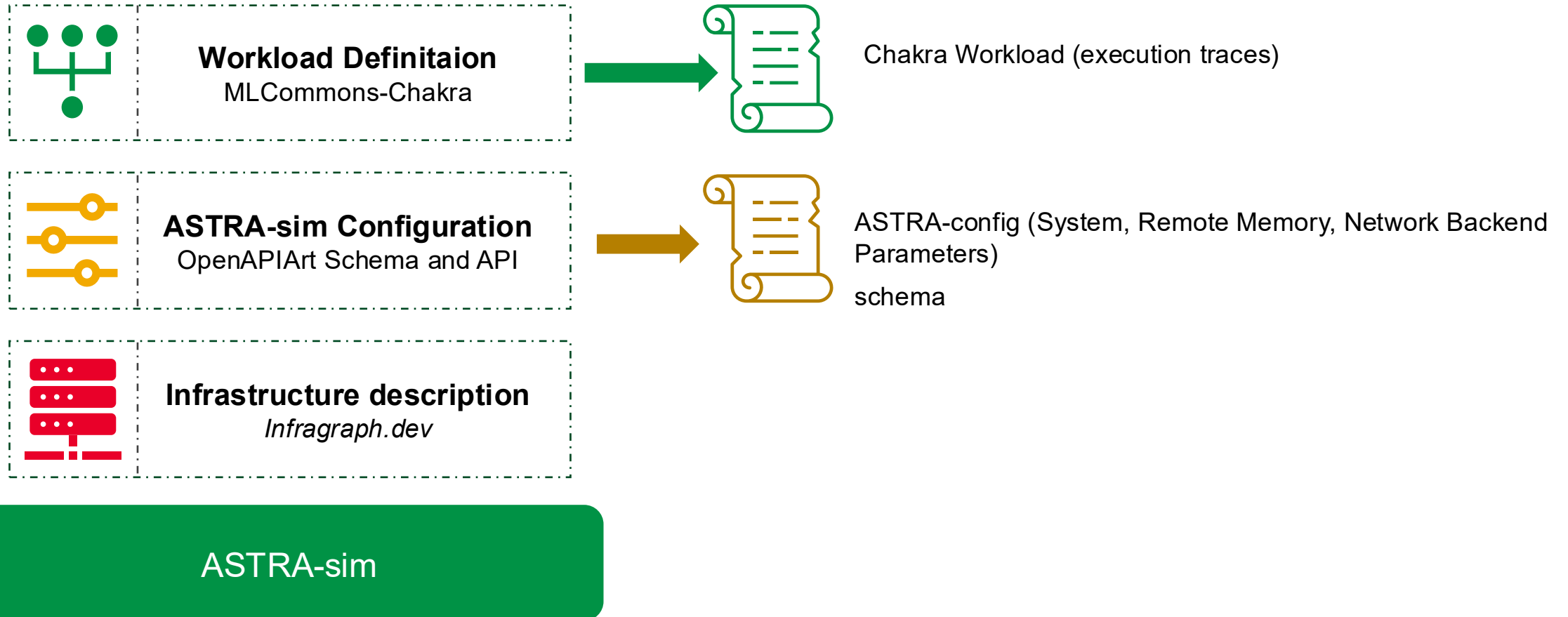
Schemas provide a common representation for these inputs



ASTRA-sim

ASTRA-sim-service

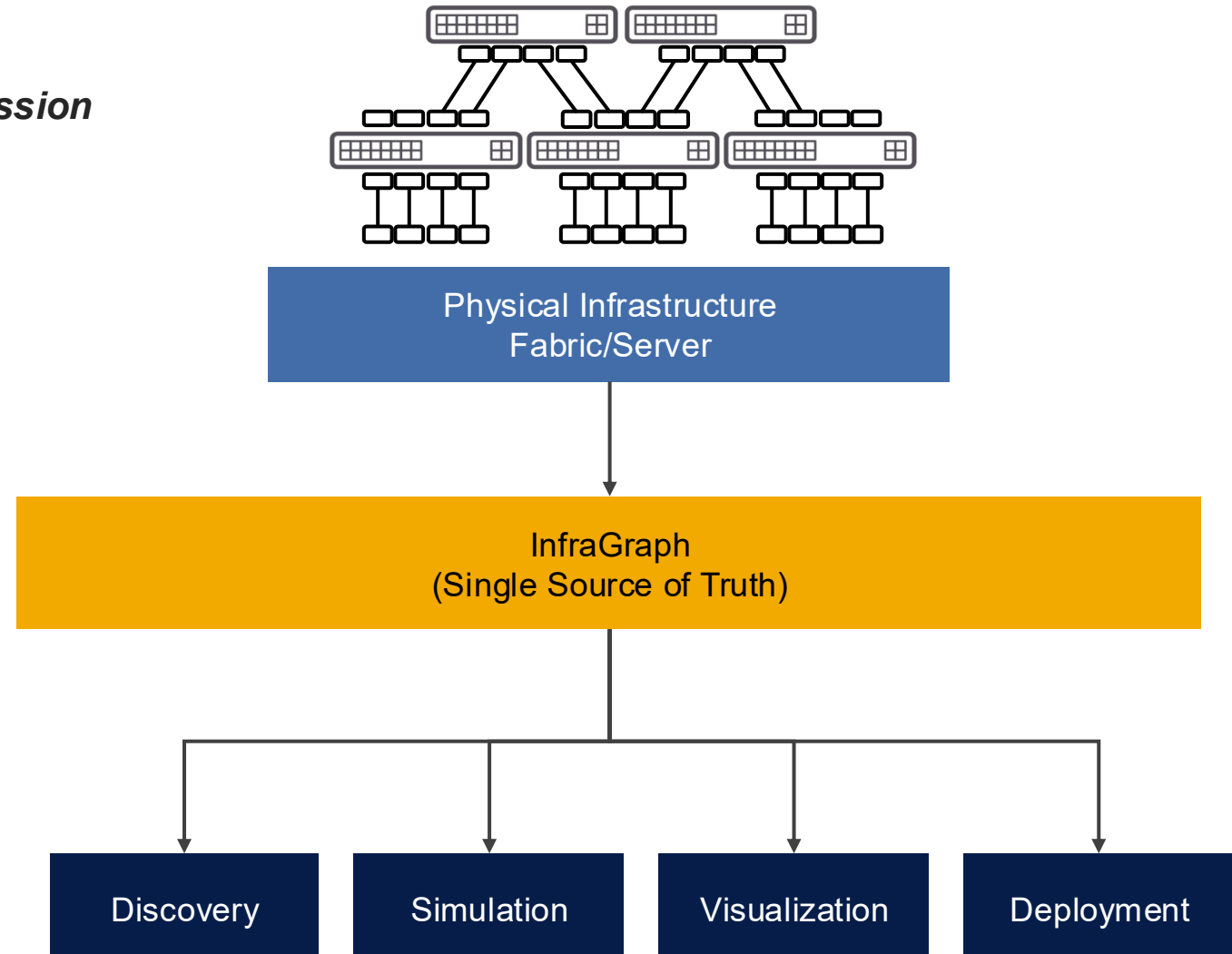
Schemas provide a common representation for these inputs



InfraGraph (Recap)

More details were covered during the morning session

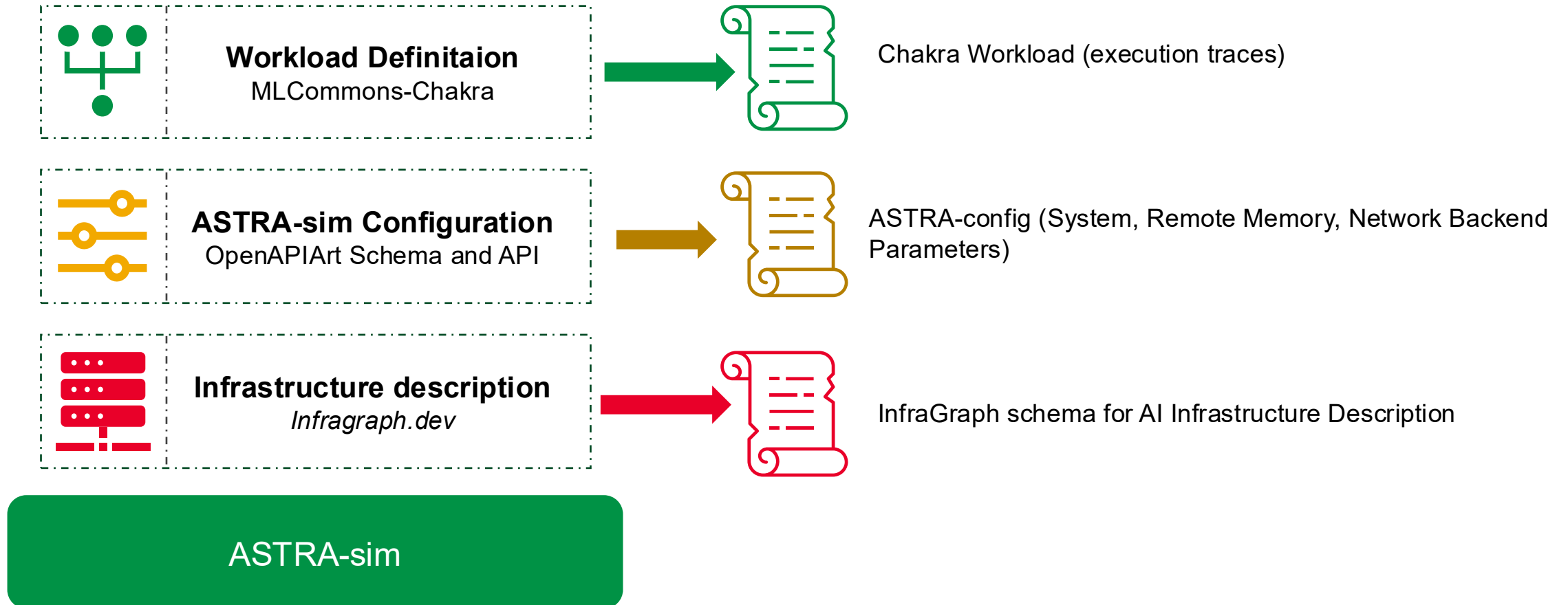
- Graph based infrastructure modelling
- Vendor neutral representation
- Workload – Infrastructure decoupling
- Cross workflow reusability
- Discover system topology
- Translate to other topology formats



A common graph representation for AI infrastructure

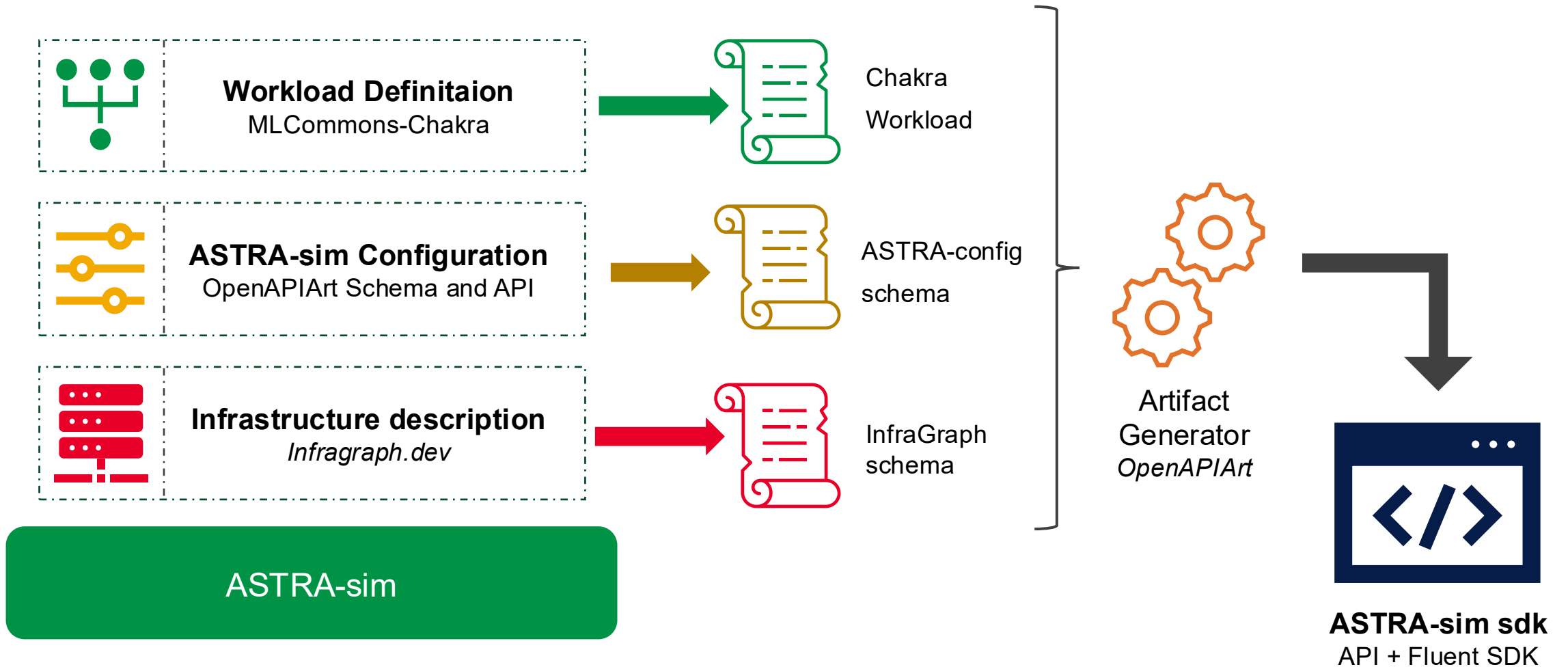
ASTRA-sim-service

Schemas provide a common representation for these inputs



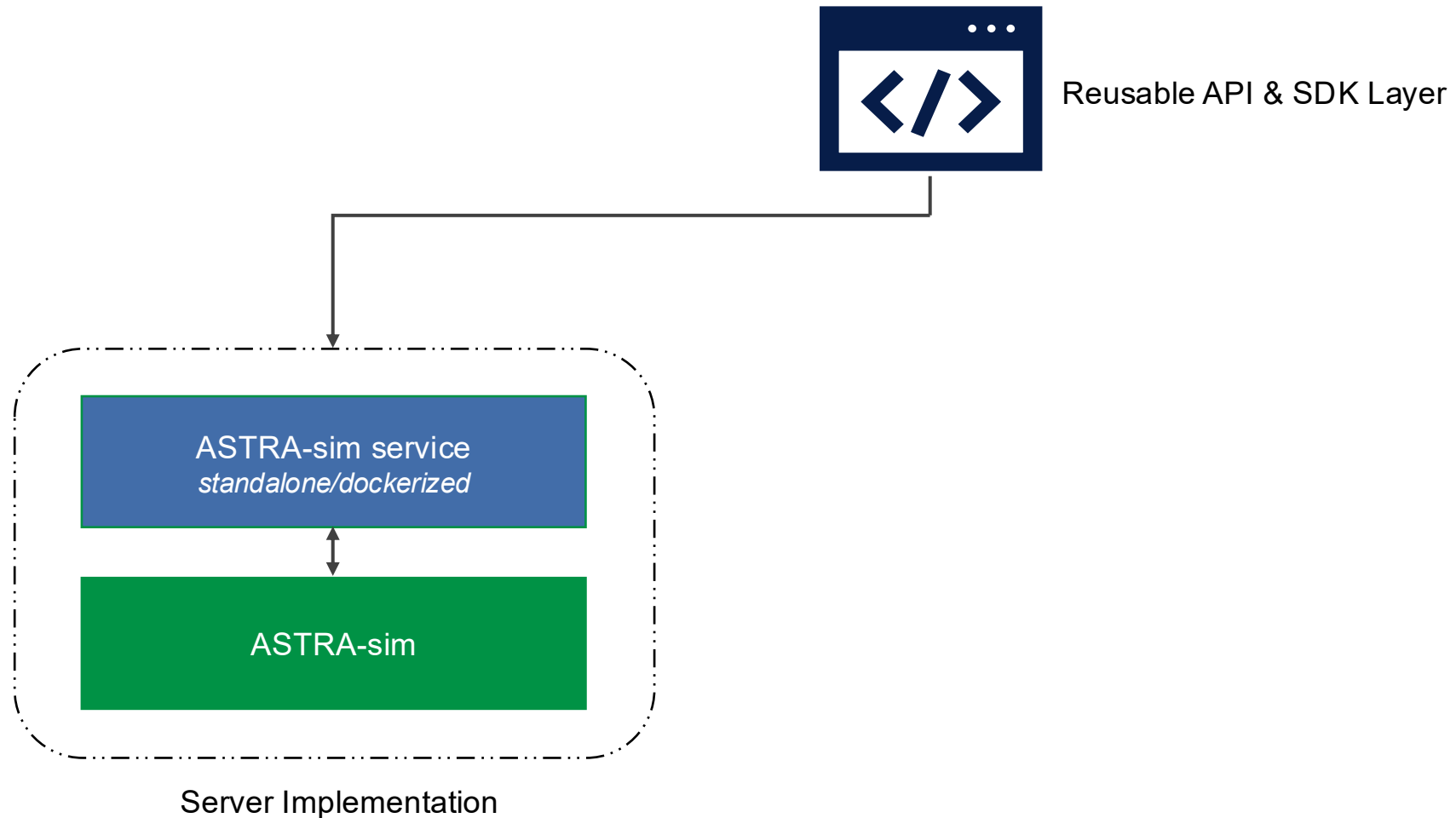
ASTRA-sim-service

Schemas provide a common representation for these inputs



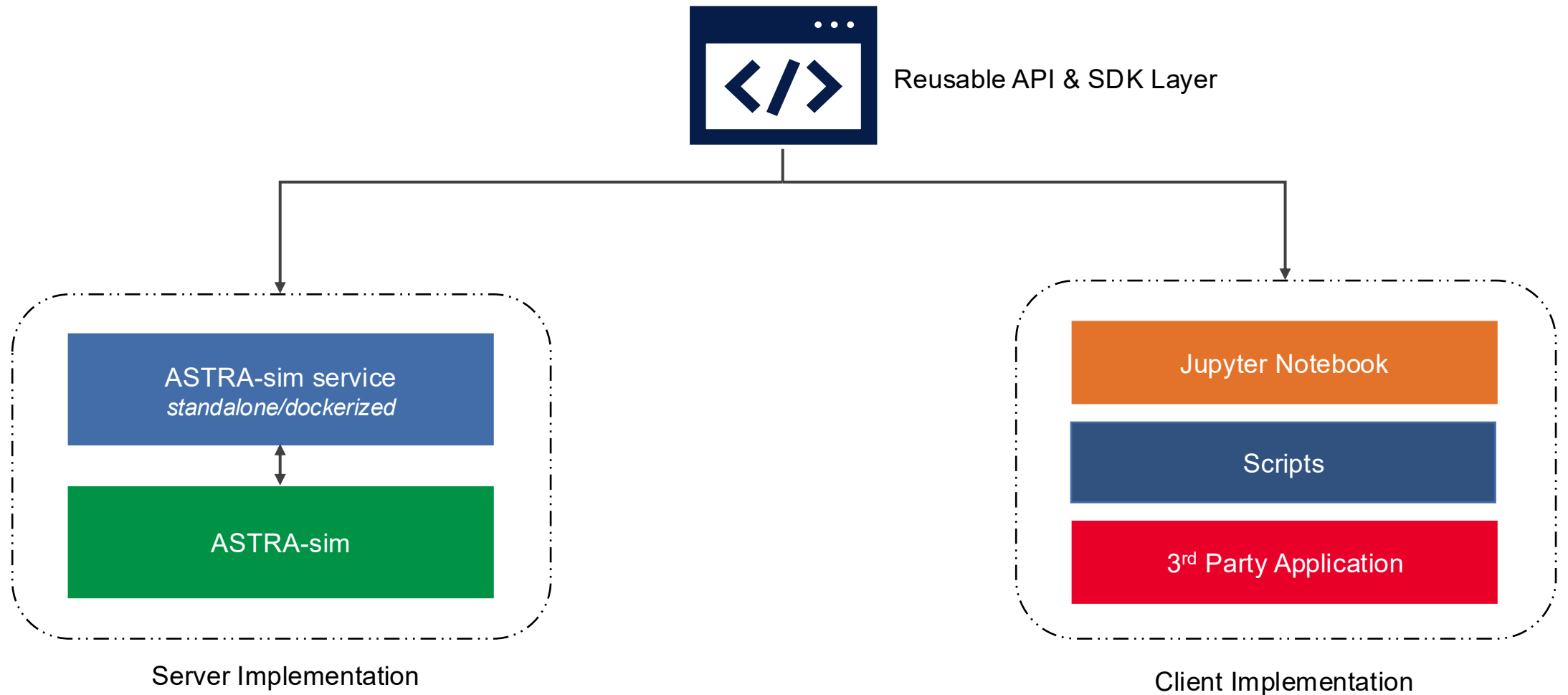
ASTRA-sim-service

The same interface supports both server and client workflows



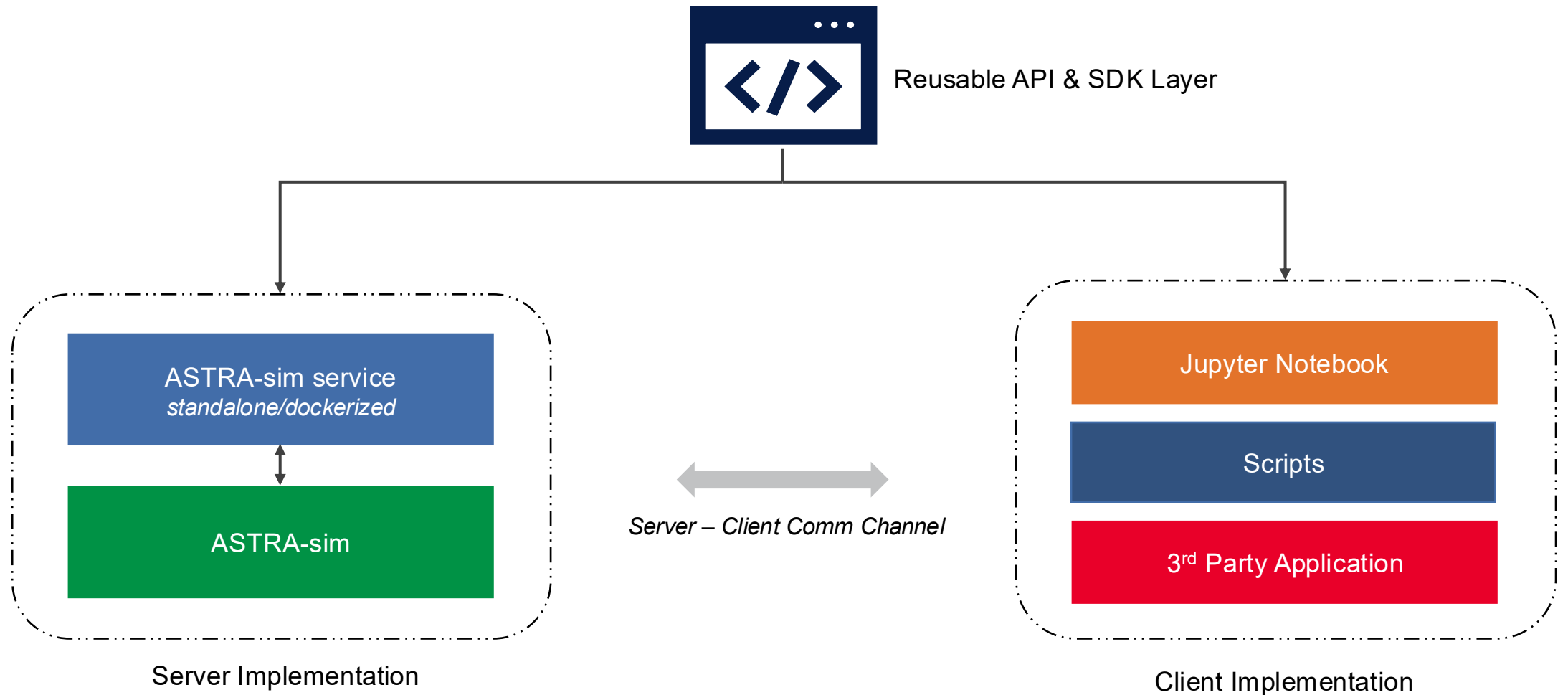
ASTRA-sim-service

The same interface supports both server and client workflows



ASTRA-sim-service

The same interface supports both server and client workflows

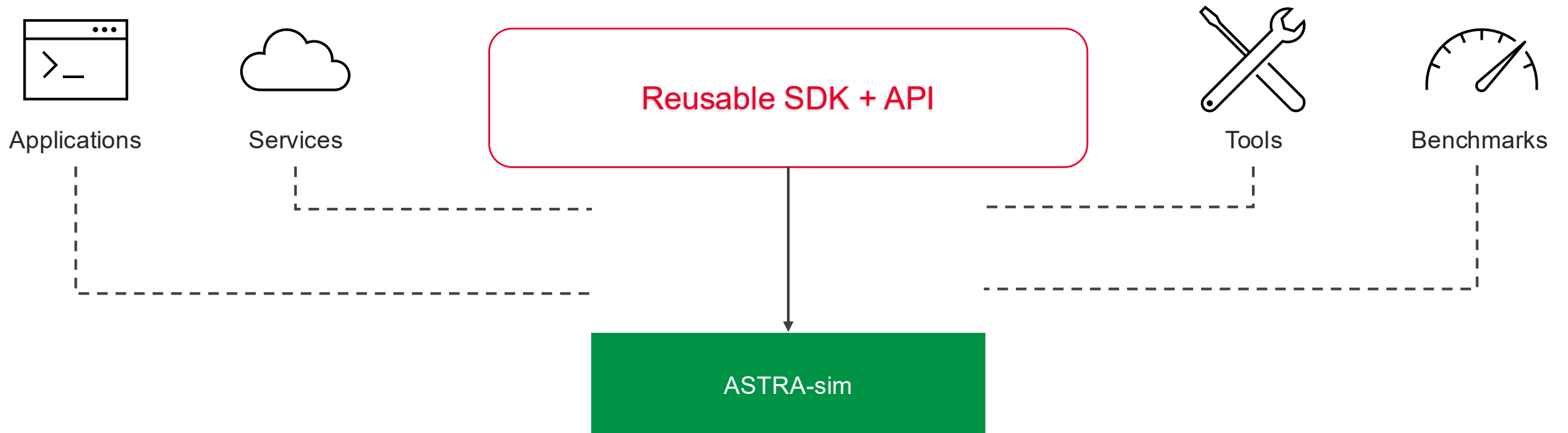


Quick Demo with ASTRA-sim-service

ASTRA-sim-service docker → Notebook & KAI DC Test Platform

Demo Part I:
ASTRA-sim-service docker
setup

Why We're Sharing This



Our goal is not to replace existing workflows, but to make simulation more accessible and reusable across them

Explore Further



ASTRA-sim-service GitHub

<https://github.com/astra-sim/astra-sim-service>



ASTRA-sim-service tutorial

<https://www.youtube.com/watch?v=07GMOhLrX84>

Thank you