

InfraGraph: Vendor-Neutral Infrastructure Topology for AI/HPC

Harsh Sikhwal
Keysight Technologies

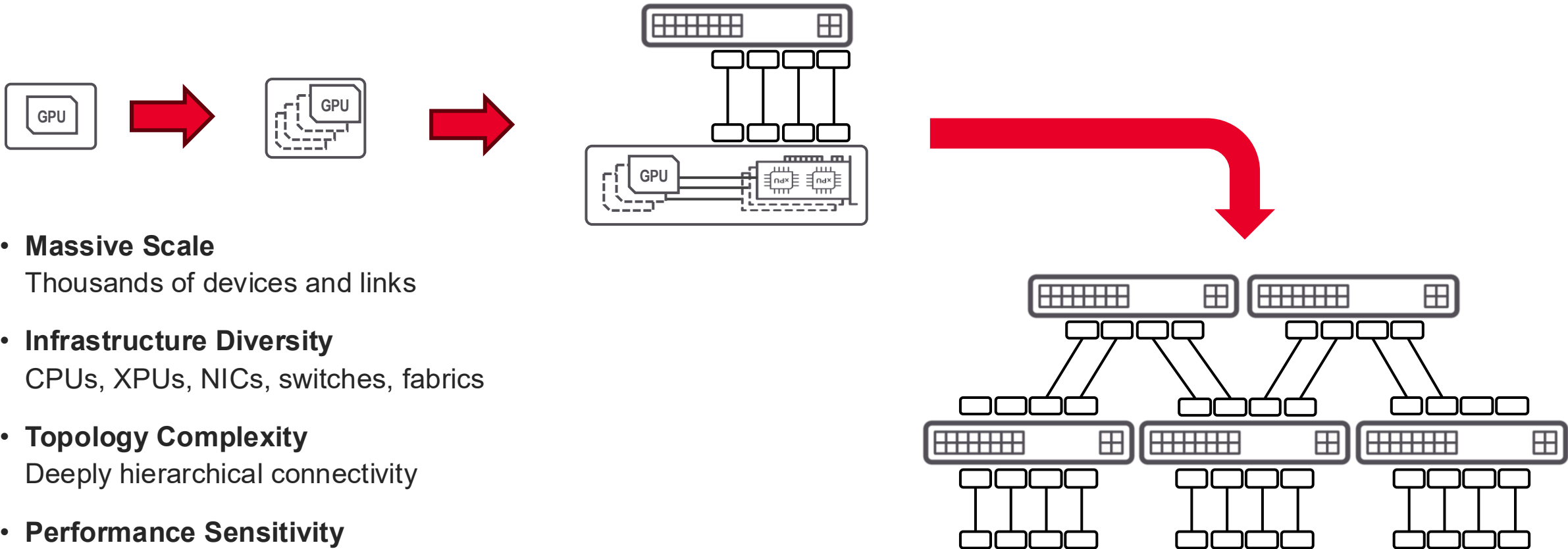
Why Infrastructure Modelling is Hard

Scale, diversity, and ecosystem fragmentation make AI infrastructure increasingly difficult to manage and reuse.



The Growing Infrastructure Modelling Problem

AI infrastructure is growing faster than our ability to describe, analyze, and reason about it



- **Massive Scale**

Thousands of devices and links

- **Infrastructure Diversity**

CPUs, XPU, NICs, switches, fabrics

- **Topology Complexity**

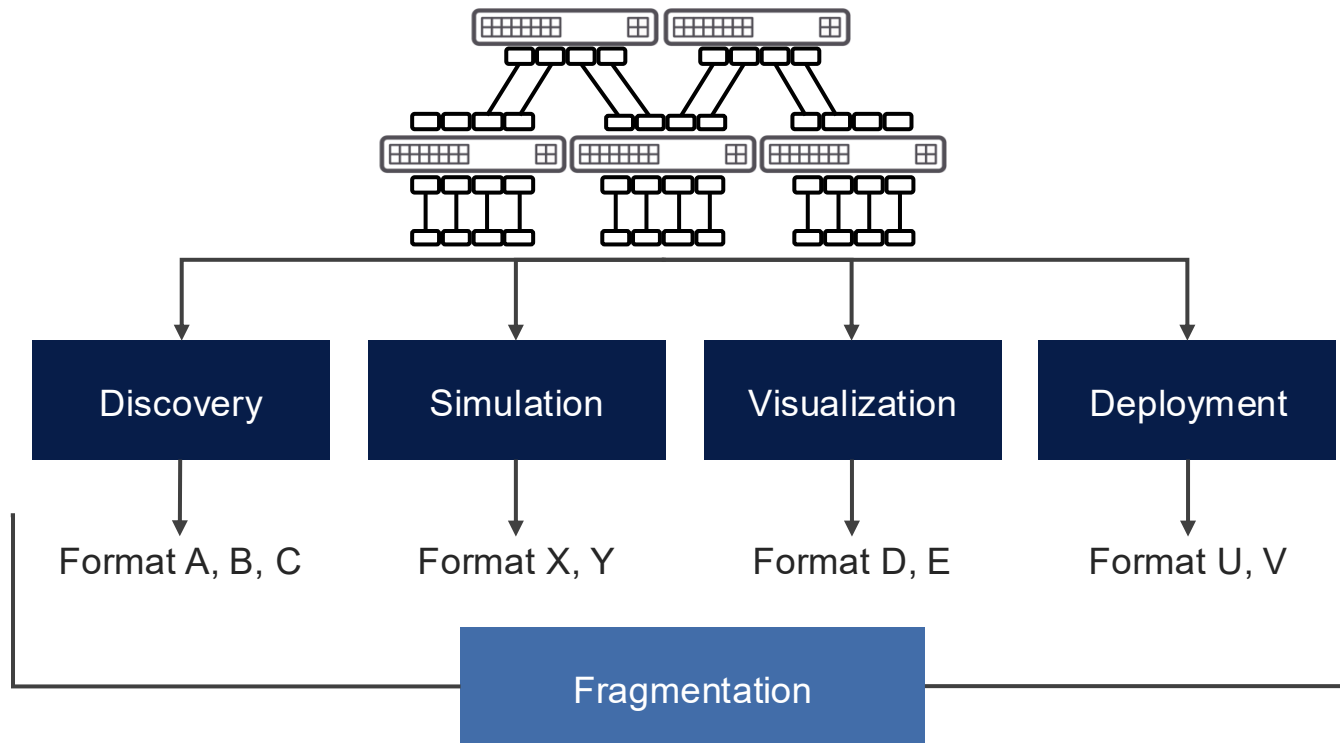
Deeply hierarchical connectivity

- **Performance Sensitivity**

Communication increasingly depends on topology

Fragmentation Across the Infrastructure Ecosystem

The same infrastructure is repeatedly recreated across disconnected tools and workflows



One infrastructure, many disconnected representations

- **Duplicate Effort**
Infrastructure recreated across workflows
- **Inconsistent Representations**
Multiple descriptions of the same system
- **Limited Interoperability**
Tools struggle to exchange information
- **Reduced Reproducibility**
Harder to share and reuse artifacts

Towards a Common Representation

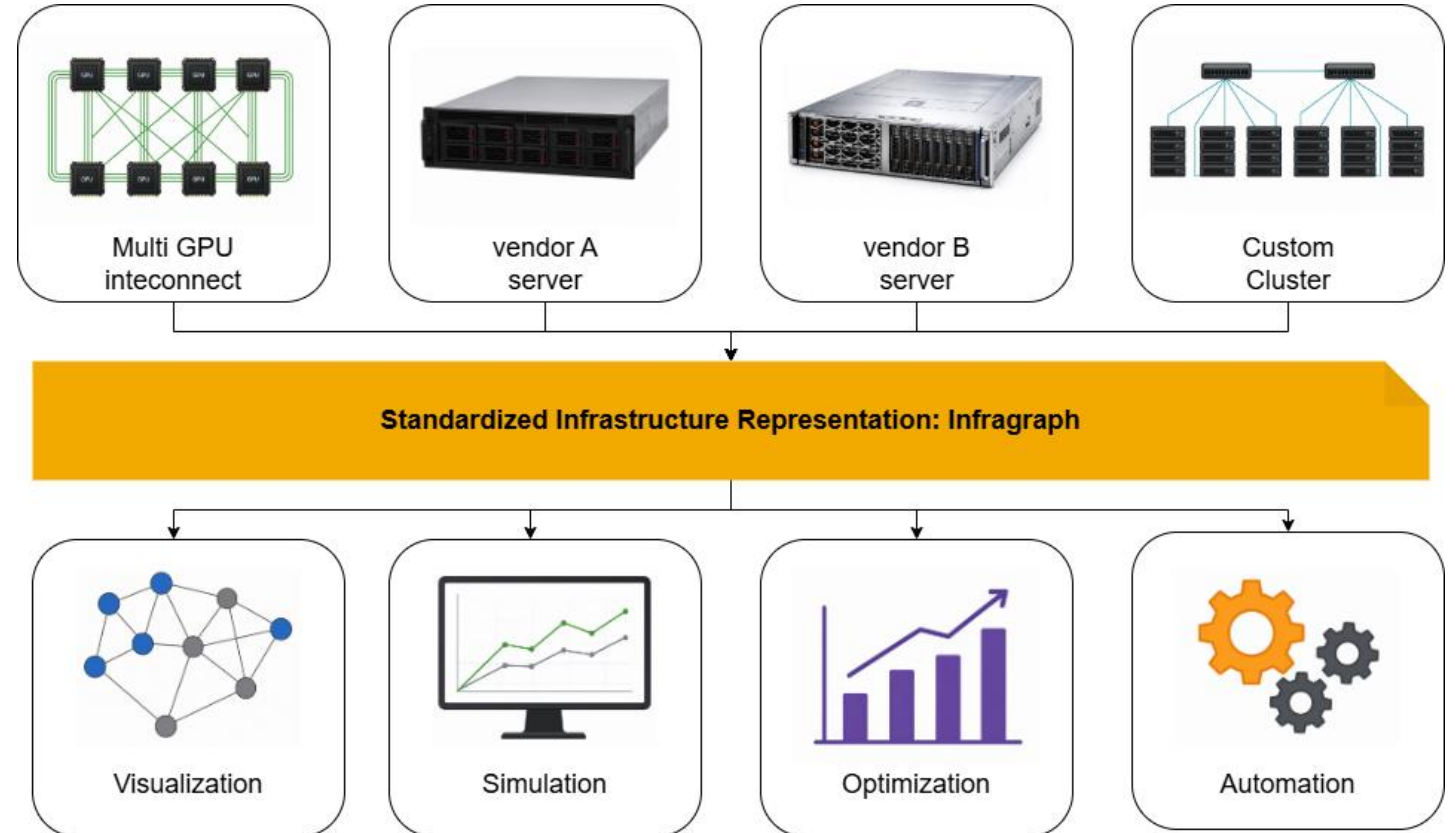
A shared representation enables interoperability, reuse, and scalable infrastructure management.



Why Standardize Infrastructure Representations?

A common representation enables interoperability, reuse, and scalable infrastructure management

- **Reusability**
Define infrastructure once
- **Interoperability**
Share descriptions across ecosystems
- **Extensibility**
Support evolving hardware and fabrics
- **Tool Independence**
Enable analysis, simulation, and visualization

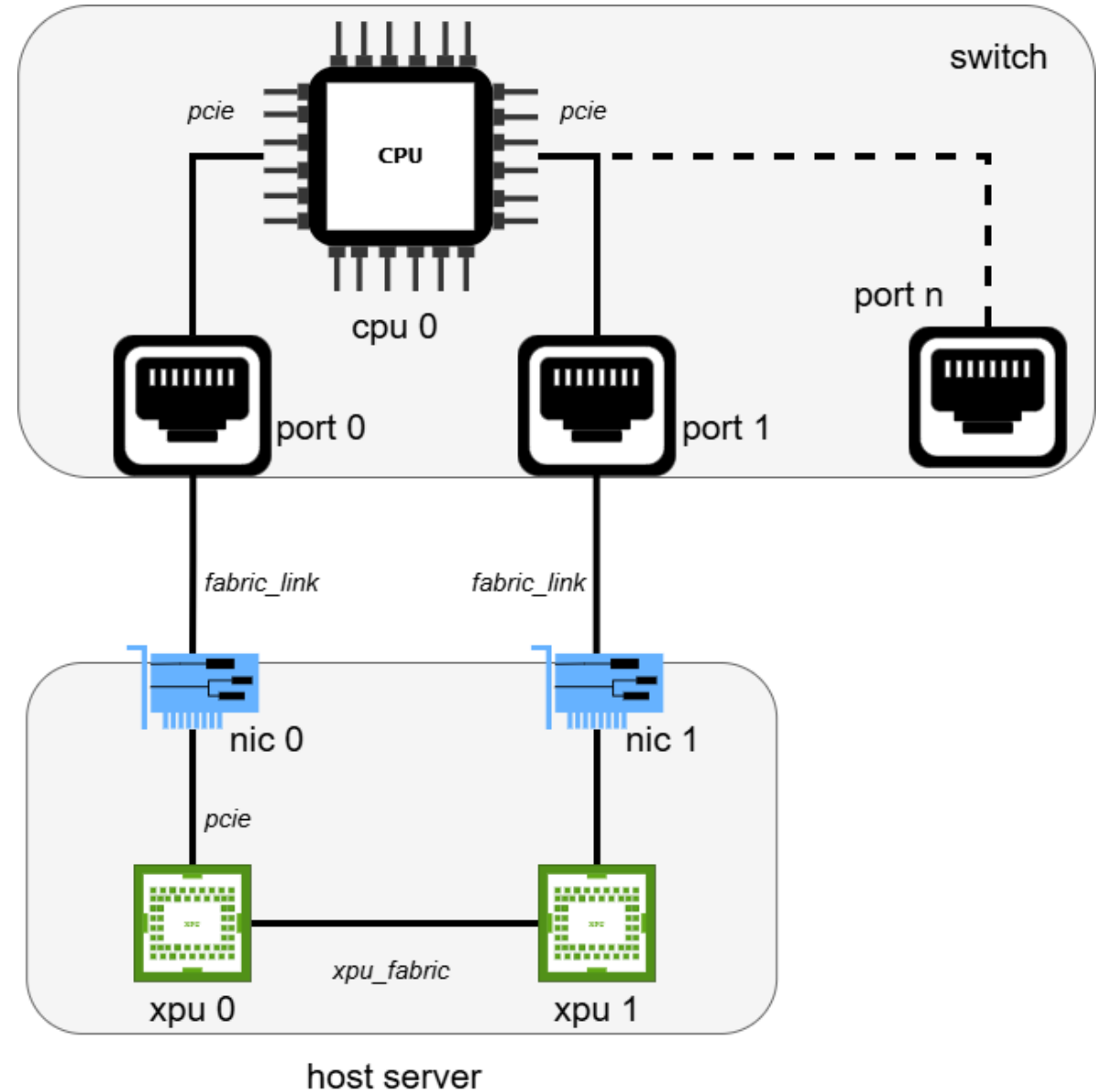


Describe infrastructure once. Reuse it everywhere.

Why a Graph is the Right Abstraction

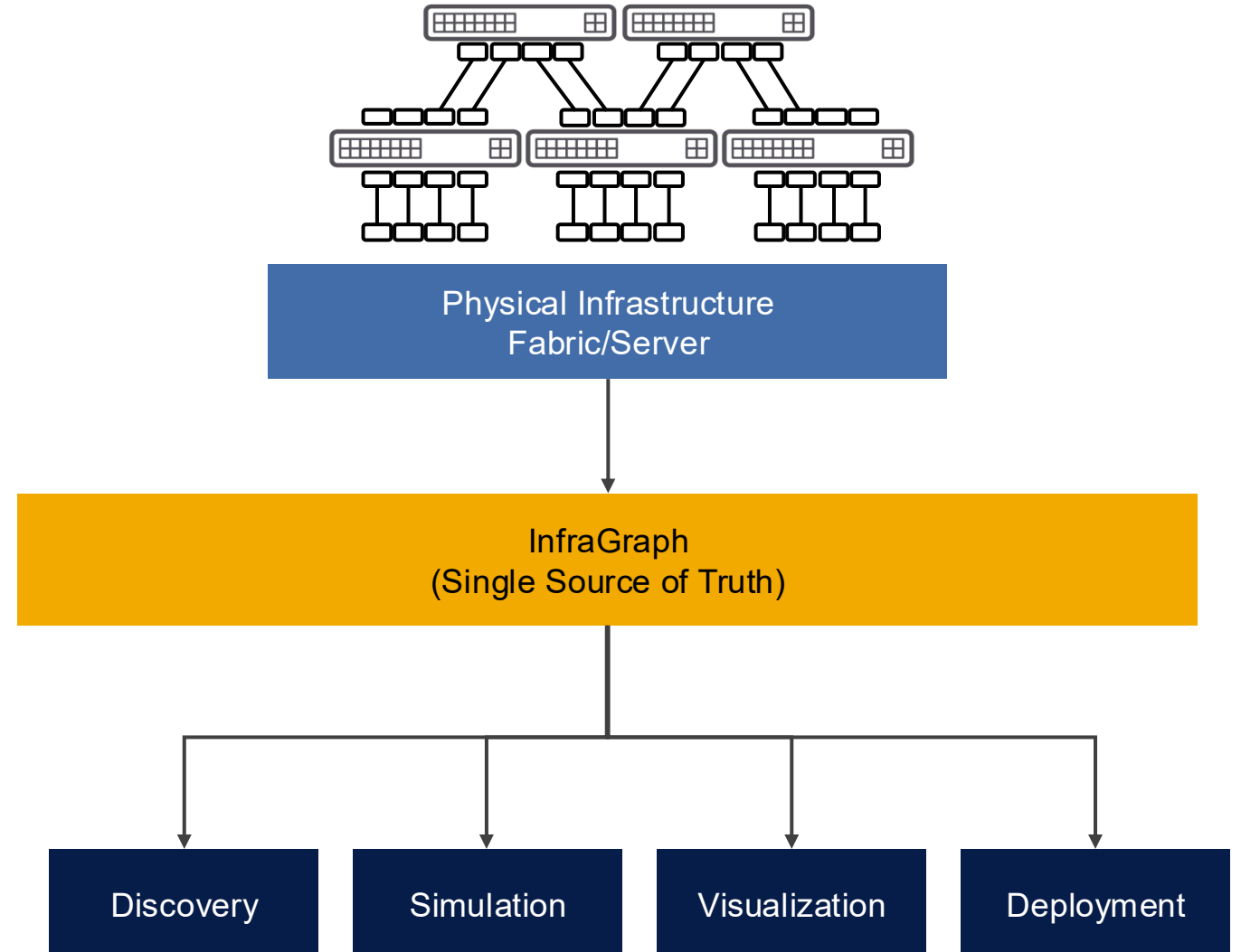
Infrastructure is naturally graph-structured

- **Infrastructure is Naturally Connected**
 - Devices and components form relationships
 - Communication depends on connectivity
- **Hierarchical by Design**
 - Components → Devices → Fabrics → Clusters
- **Metadata Rich**
 - Nodes and links carry attributes
 - Bandwidth, latency, placement, policies
- **Extensible and Queryable**
 - Support evolving hardware and topologies



Introducing InfraGraph

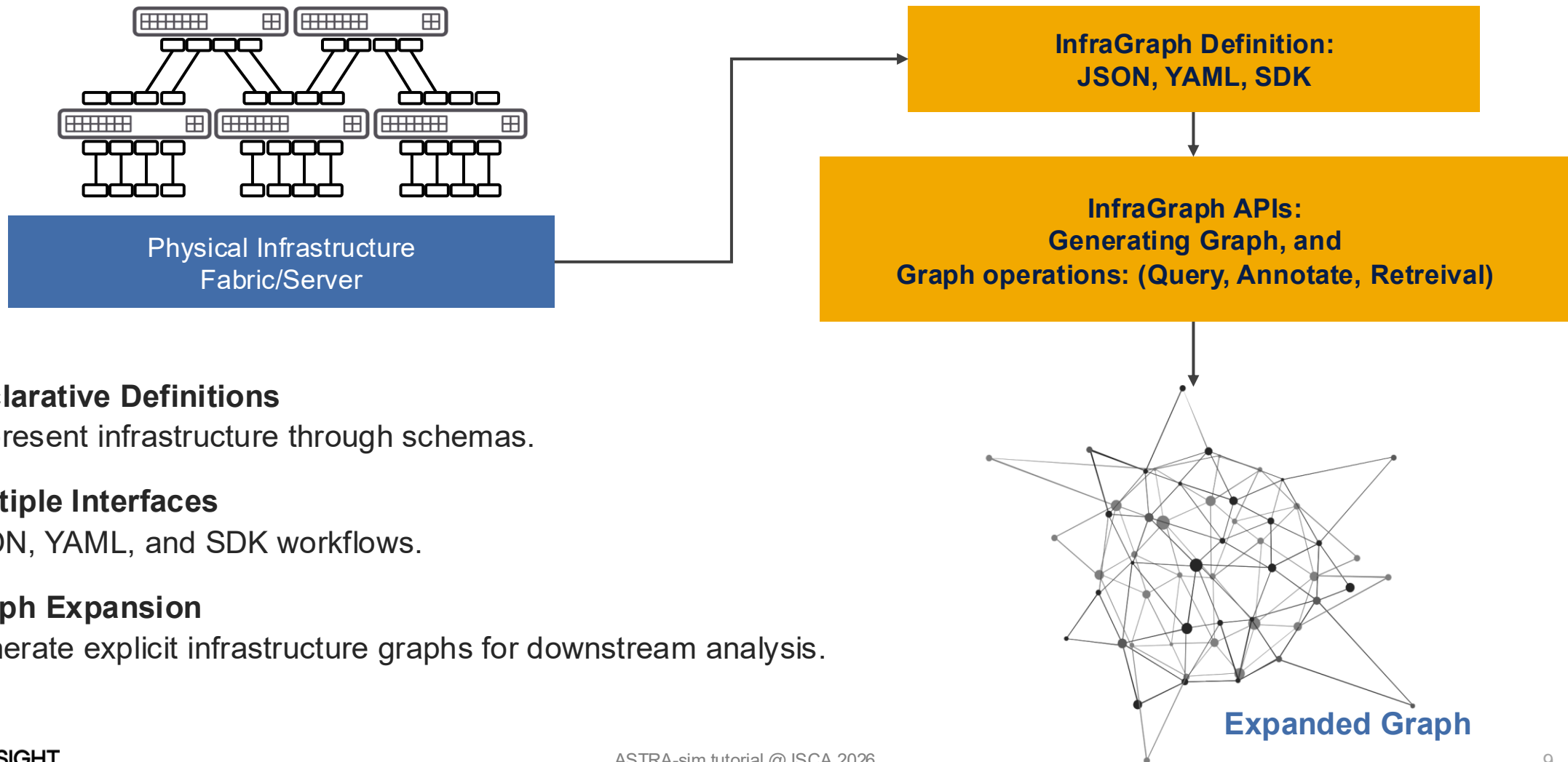
InfraGraph provides a vendor-neutral schema and API for describing AI infrastructure as graphs, enabling analysis, simulation, optimization, and interoperability across tools.



A common graph representation for AI infrastructure.

Declarative Infrastructure Modelling

Define infrastructure once through schemas, APIs, and graph expansion.

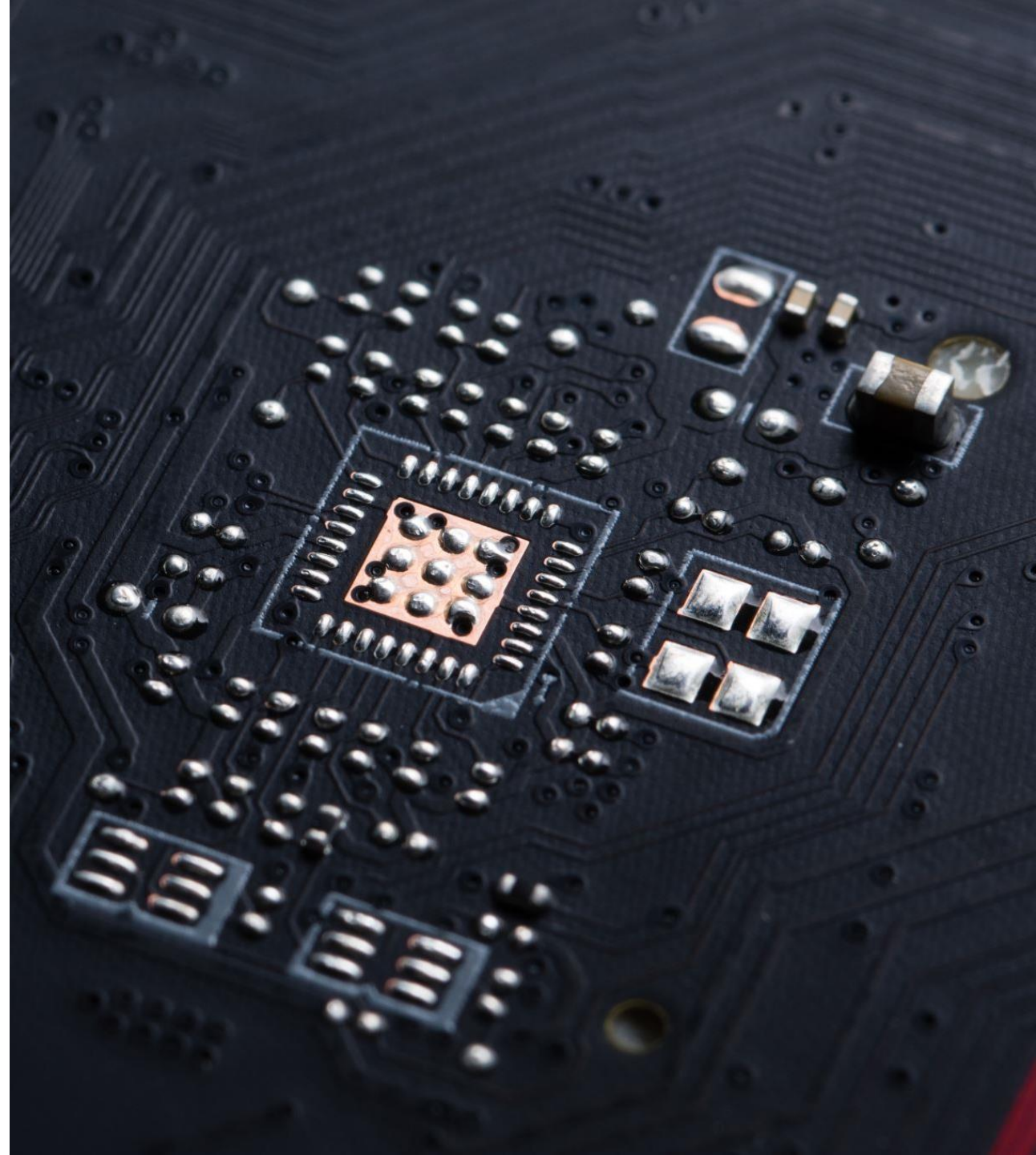


- **Declarative Definitions**
Represent infrastructure through schemas.
- **Multiple Interfaces**
JSON, YAML, and SDK workflows.
- **Graph Expansion**
Generate explicit infrastructure graphs for downstream analysis.

Building Infrastructure from Reusable Components

Building an Intra-Device

We start from the smallest unit of infrastructure — a device. Components such as CPUs, XPU, and switches are modeled as graph entities and relationships.



AI Infrastructure Device Representation

Modeling an AI infrastructure through reusable components, link types, and connectivity.

```
devices:  
  name:
```

Device Definition
Reusable blueprint



server

AI Infrastructure Device Representation

Modeling an AI infrastructure through reusable components, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:
```

Device Definition

Reusable blueprint

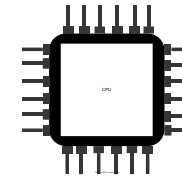


server

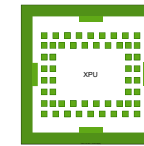
1

Components

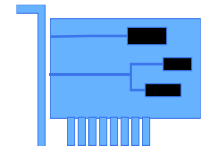
Device internals – name, count & type



cpu – count 2



xpu – count 8



nic – count 8

AI Infrastructure Device Representation

Modeling an AI infrastructure through reusable components, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:  
  links:  
    - name:
```

Device Definition

Reusable blueprint

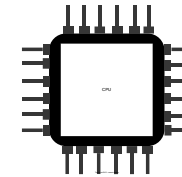


server

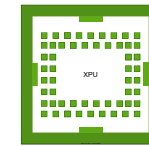
1

Components

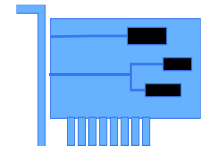
Device internals – name, count & type



cpu – count 2



xpu – count 8



nic – count 8

2

Link Definition

Communication characteristics



xpu fabric – 1600 Gbps

AI Infrastructure Device Representation

Modeling an AI infrastructure through reusable components, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:  
  links:  
    - name:  
  edges:  
    - ep1 - link_name - ep2
```

Device Definition

Reusable blueprint

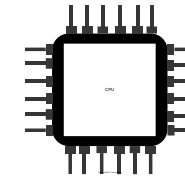


server

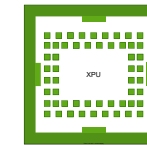
1

Components

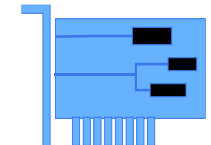
Device internals – name, count & type



cpu – count 2



xpu – count 8



nic – count 8

2

Link Definition

Communication characteristics

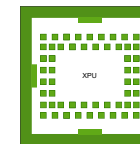


xpu fabric – 1600 Gbps

3

Edge Creation

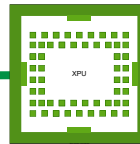
Connecting component instances with defined links



xpu[0:m]



xpu fabric



xpu[0:n]

Scale-Out Fabric Graph

We zoom outward from the device. Servers, switches, and interconnects are modeled as graph entities and relationships, capturing the structure of racks, pods, and AI clusters.



AI Fabric Representation

Modeling an AI infrastructure through reusable devices, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:  
  links:  
    - name:  
  edges:  
    - ep1 - link_name - ep2
```

1

Device Definition

Reusable blueprint



server



switch

AI Fabric Representation

Modeling an AI infrastructure through reusable devices, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:  
  links:  
    - name:  
  edges:  
    - ep1 - link_name - ep2  
instances:  
  - name:  
    instance_name:  
    count:
```

1

Device Definition

Reusable blueprint



server



switch

2

Instances

Copies of defined device



switch 0



server 0

....



server n

AI Fabric Representation

Modeling an AI infrastructure through reusable devices, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:  
  links:  
    - name:  
  edges:  
    - ep1 - link_name - ep2  
instances:  
  - name:  
    instance_name:  
    count:  
links:  
  - name
```

1

Device Definition

Reusable blueprint



server



switch

2

Instances

Copies of defined device



switch 0



server 0

....



server n

3

Link Definition

Communication characteristics



ethernet_link – 100 Gbps

AI Fabric Representation

Modeling an AI infrastructure through reusable devices, link types, and connectivity.

```
devices:  
  name:  
  components:  
    - name:  
      type:  
      count:  
  links:  
    - name:  
  edges:  
    - ep1 - link_name - ep2  
instances:  
  - name:  
    instance_name:  
    count:  
links:  
  - name  
edges:  
  - ep1 - link_name - ep2
```

1

Device Definition

Reusable blueprint



server



switch

2

Instances

Copies of defined device



switch 0



server 0

...



server n

3

Link Definition

Communication characteristics



ethernet_link - 100 Gbps

4

Edge Creation

Connecting instances with defined links



switch 0

port 0

ethernet_link



server 0

Infrastructure as a Graph

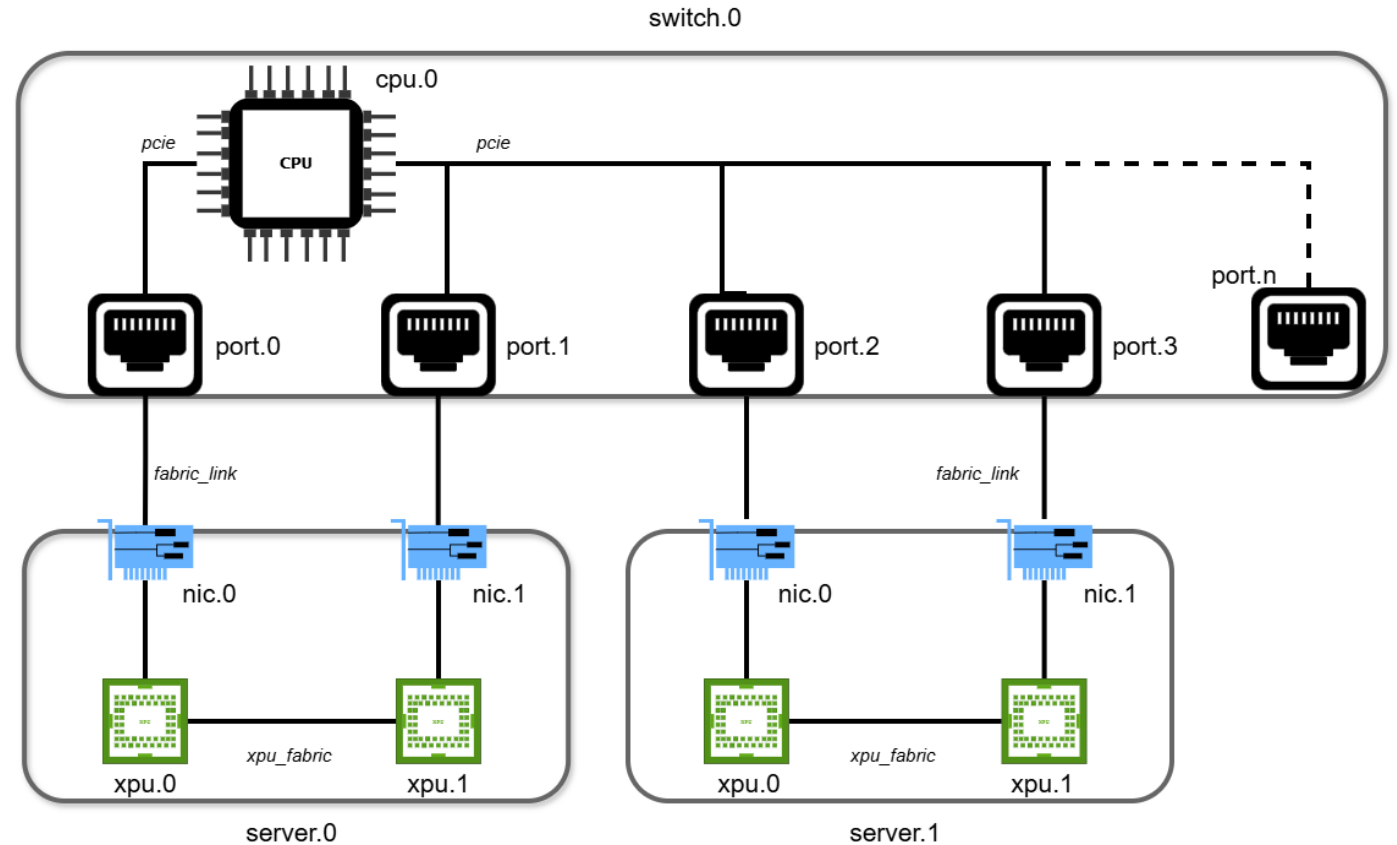
From reusable infrastructure descriptions to explicit graph representations. Components become nodes, connectivity becomes edges, and hierarchy is preserved through graph abstractions.



Infrastructure as a Graph

From reusable abstractions to graph entities and connectivity

- Devices composed of components
- Components are lowest level
- Components become graph nodes
- Connected components are edges
- **Principle of Instances**
 - Components can be replicated
 - Devices can be replicated



Graph Expansion

Transforming infrastructure descriptions into an explicit graph representation.

- **Graph Node Creation:**

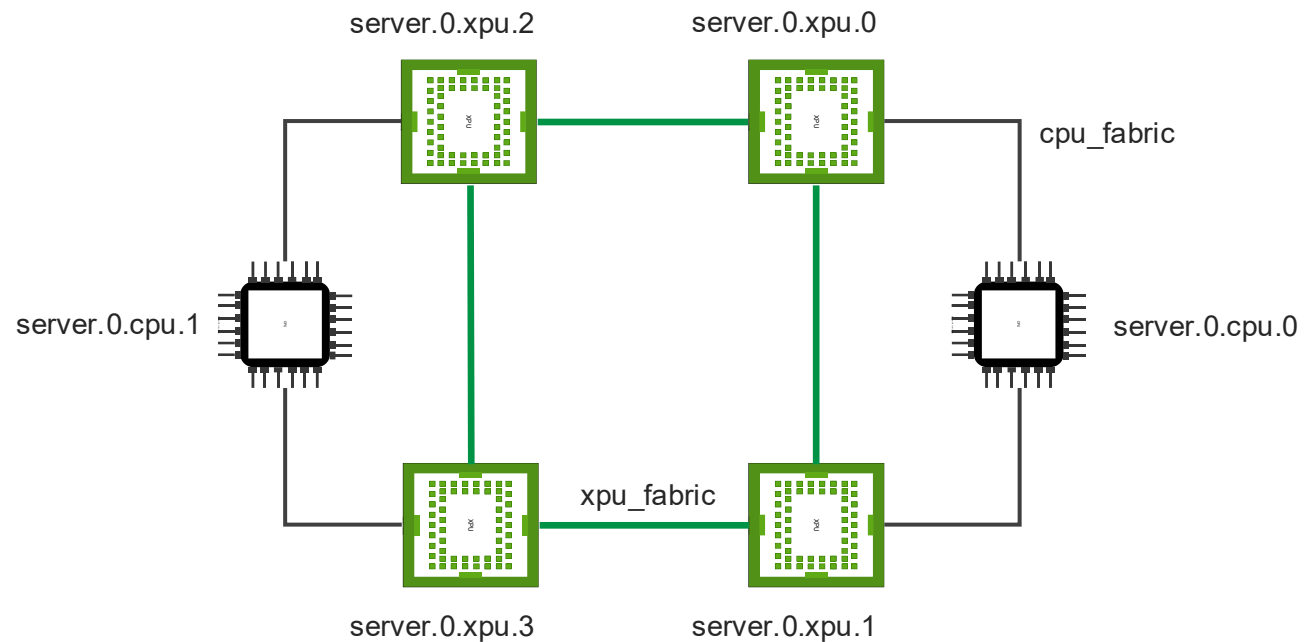
- Define device
- Instantiate devices
- Instantiate device components:
 - *server.0.cpu.0*
 - *server.0.cpu.1*
 - ...
 - *server.1.xpu.7*

- **Graph Node representation:**

- *device.index.component.index*

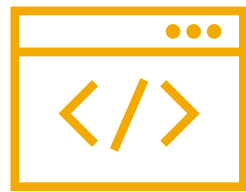
- **Edge Representation**

- *Ep1: device.index.component.index*
- *Ep2: device.index.component.index*
- *Link: Link name*



InfraGraph APIs

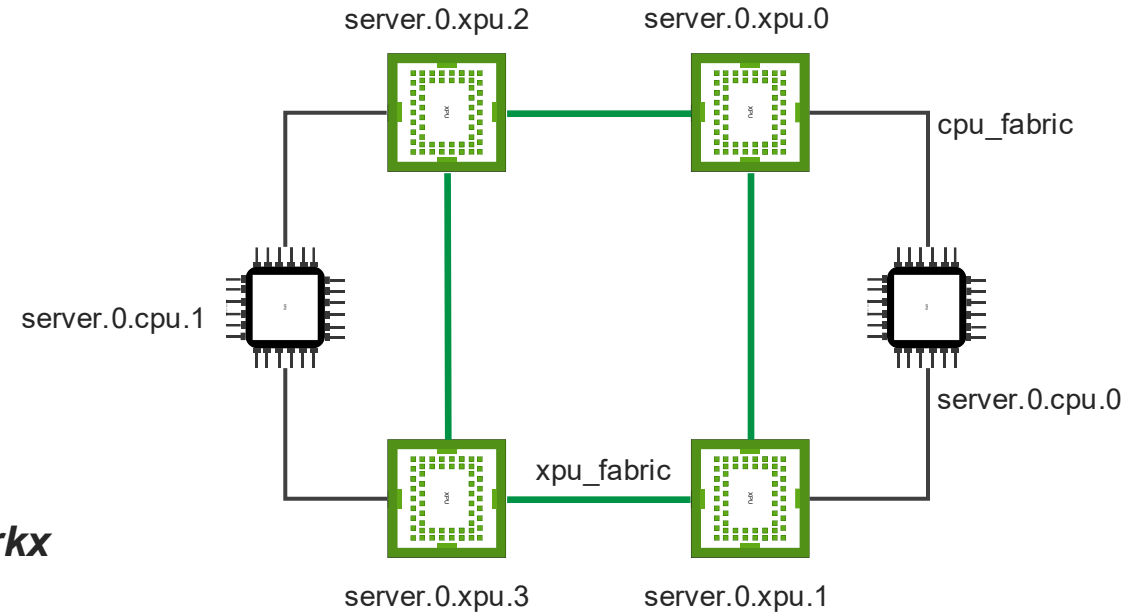
Programmatic interfaces for graph generation, annotation, retrieval, and querying



InfraGraph Definition

InfraGraph APIs

- Expansion to Graph - *networkx*
- Annotation
- Query
- Retrieval:
(*json, yaml, infrastructure, networkx*)

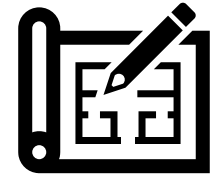


InfraGraph Tooling

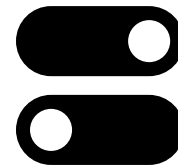
Discover, model, translate, and visualize infrastructure through a unified tooling ecosystem built around InfraGraph.



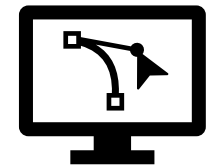
**Discover System/Network
Topology**



**Reusable Infrastructure
Templates**



**Translate topologies to
and from InfraGraph**



**Interactive infrastructure
exploration**

Quick Demo

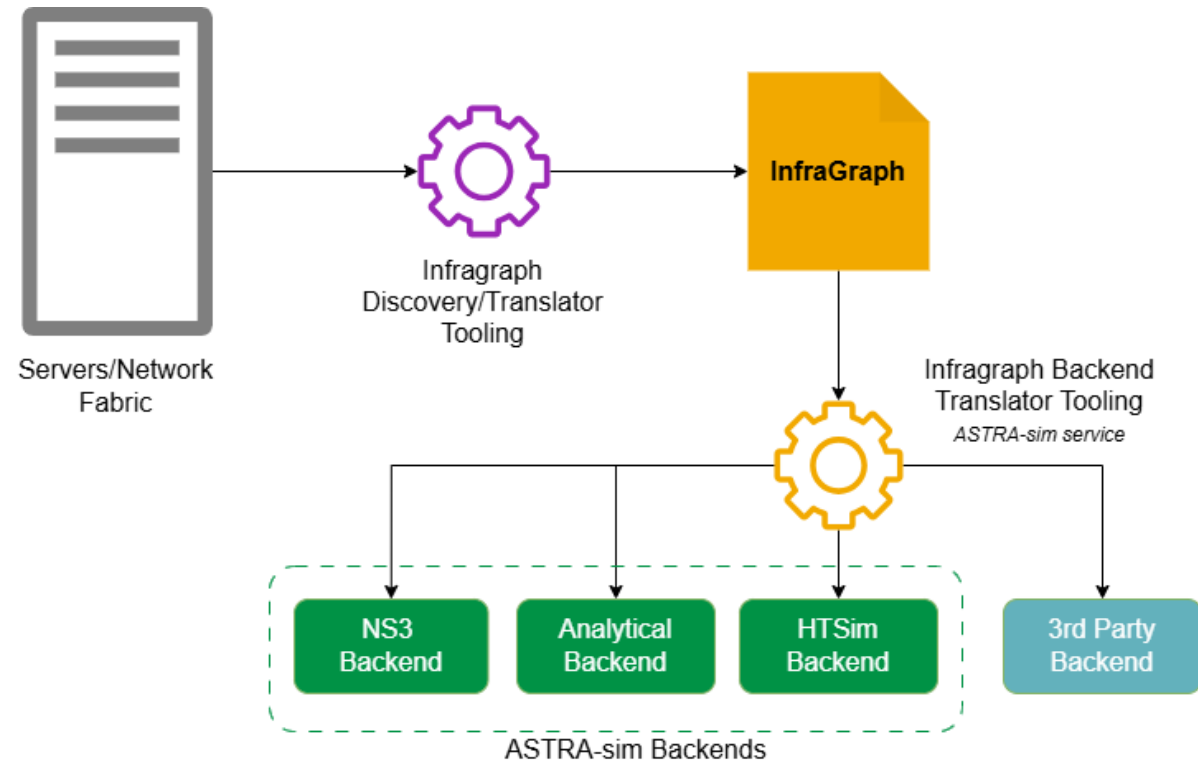
InfraGraph Tooling → Translate and Visualize

(venv) ixia@ixia-2025:~/demo\$ |

InfraGraph → ASTRA-sim Integration

Leveraging a common infrastructure representation across simulation backends.

- **Single Infrastructure Source of Truth**
Model servers, accelerators, networks, and fabrics once in InfraGraph.
- **Automated Topology Generation**
Translate infrastructure graphs directly into Astra-Sim backend topologies.
- **Backend-Agnostic Infrastructure Modeling**
Reuse the same infrastructure description across HTSim, NS3, analytical, and future backends.
- **Improved Reproducibility and Consistency**
Eliminate duplicated topology definitions and ensure consistent simulation inputs.



Quick Demo

InfraGraph Blueprint → Analytical and NS3 Topology

EXPLORER

OPEN EDITORS

INFRAGRAPH [WSL: UBUNTU-22.04]

- > .github
- > .venv
- > .vscode
- > api
- > artifacts
- > css
- > dist
- > docs
- > js
- > models
- > src
- > svg_images
- > .gitignore
- generate.py
- LICENSE
- Makefile
- pyproject.toml
- README.md
- requirements.txt

OUTLINE

TIMELINE



Open Chat Ctrl + Alt + I

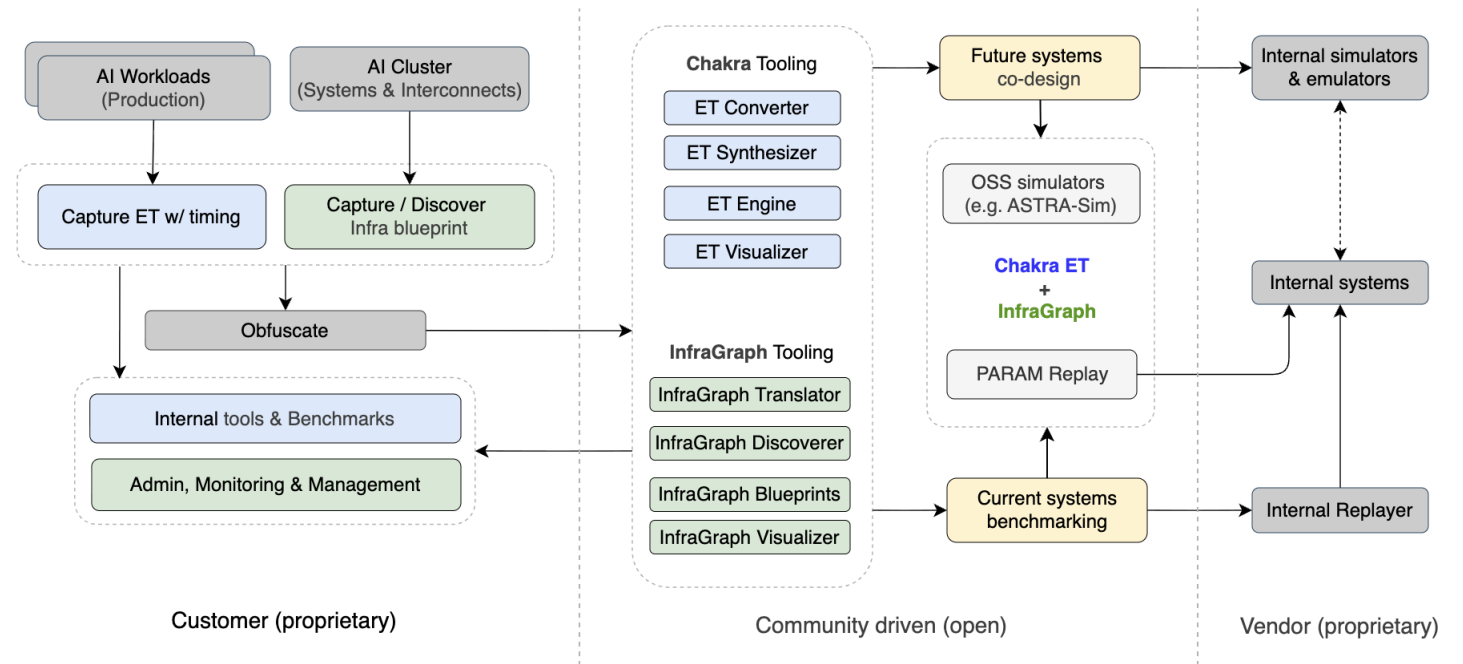
Show All Commands Ctrl + Shift + P

Find in Files Ctrl + Shift + F

Chakra and InfraGraph Ecosystem

More than a schema — an ecosystem for discovering, modeling, visualizing, and sharing AI/HPC infrastructure

- Workload + Infrastructure pairing
- Artifact sharing and reproducibility
- System-level simulation and co-design
- ASTRA-sim integration



Explore Further



InfraGraph GitHub

<https://github.com/Keysight/infragraph/>



ASTRA-sim-service GitHub

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

Thank you