



Institute for the Wireless Internet of Things

at Northeastern University

GPUs Overview

Davide Villa



Platforms for Advanced
Wireless Research



MITRE



MASSACHUSETTS
TECHNOLOGY
COLLABORATIVE



N COLOSSEUM
at Northeastern University

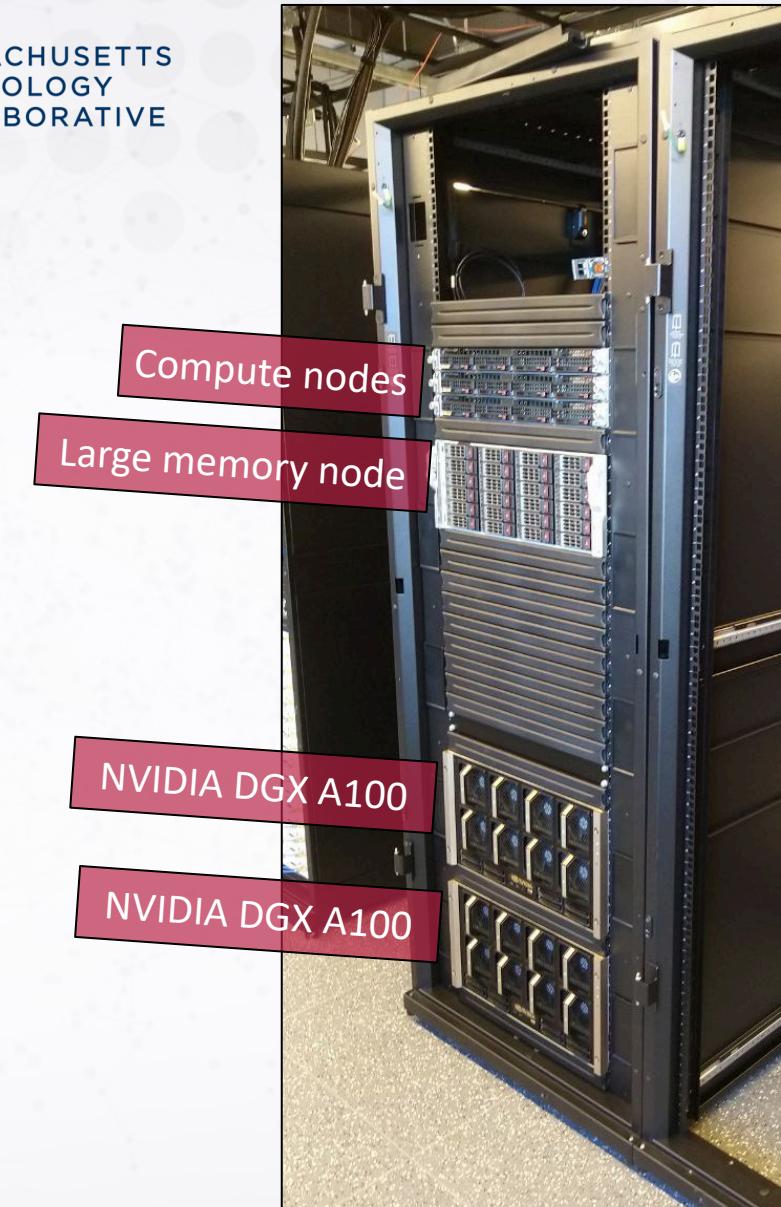
AI Jumpstart Integration



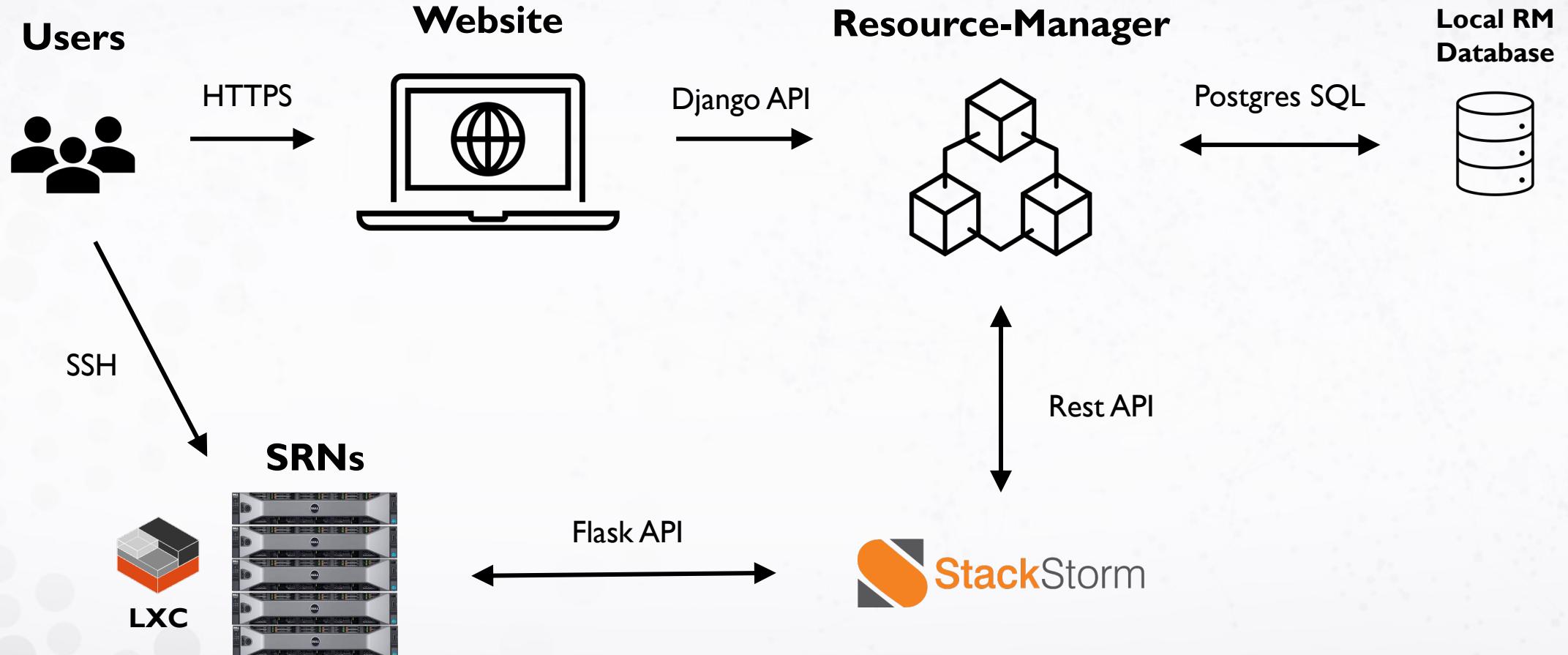
- Build GPU-accelerated software-defined, cloud-native applications for the 5G vRAN
 - 2x NVIDIA DGX A100
 - 1x Large memory node
 - 3x compute nodes
 - 1x MLX Infiniband switch



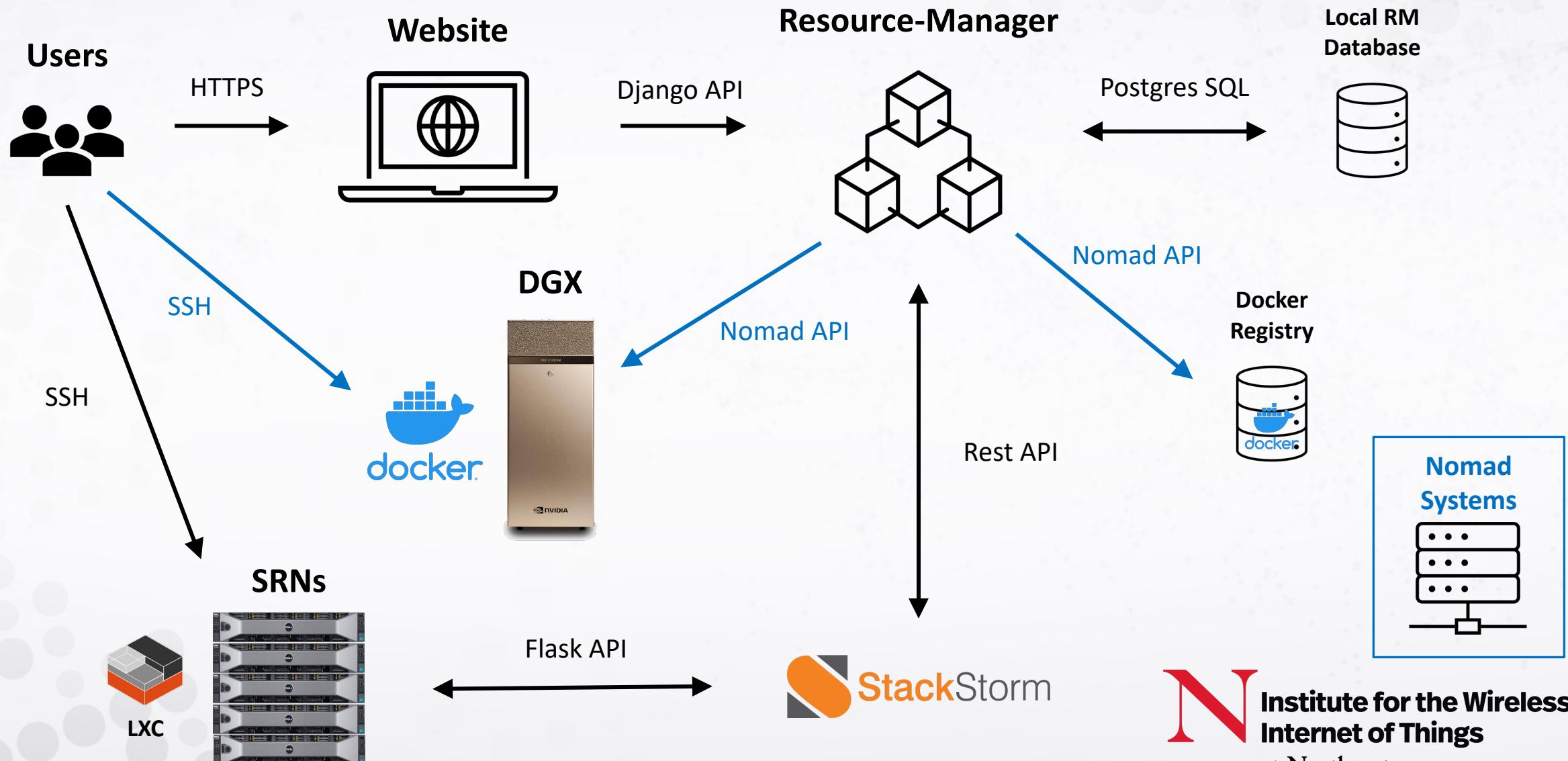
How do we **integrate** the new rack into Colosseum?



Current SRN reservation flow diagram



New GPU Reservation Flow Diagram



Nomad: Overview

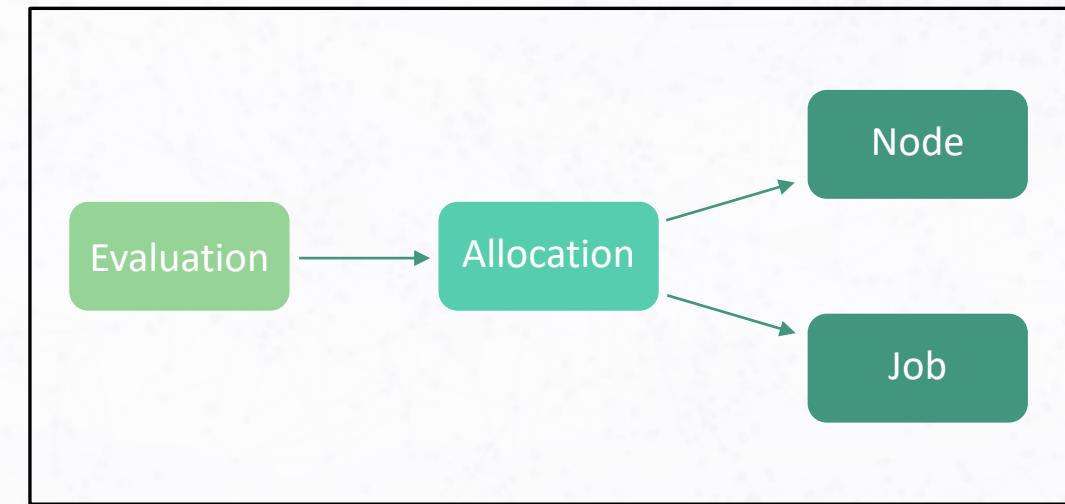


- Workload **orchestrator** to deploy and manage containers.
- Built-in support for **GPU** workloads and device plugins.
- **HashiCorp** ecosystem (Vault, Consul...).
- Compare to the more popular  **kubernetes**
 - Smaller scope.
 - More simple and lightweight.
 - More scalable (5k vs 10k nodes).
 - Flexible workload support.
 - Consistent deployment.



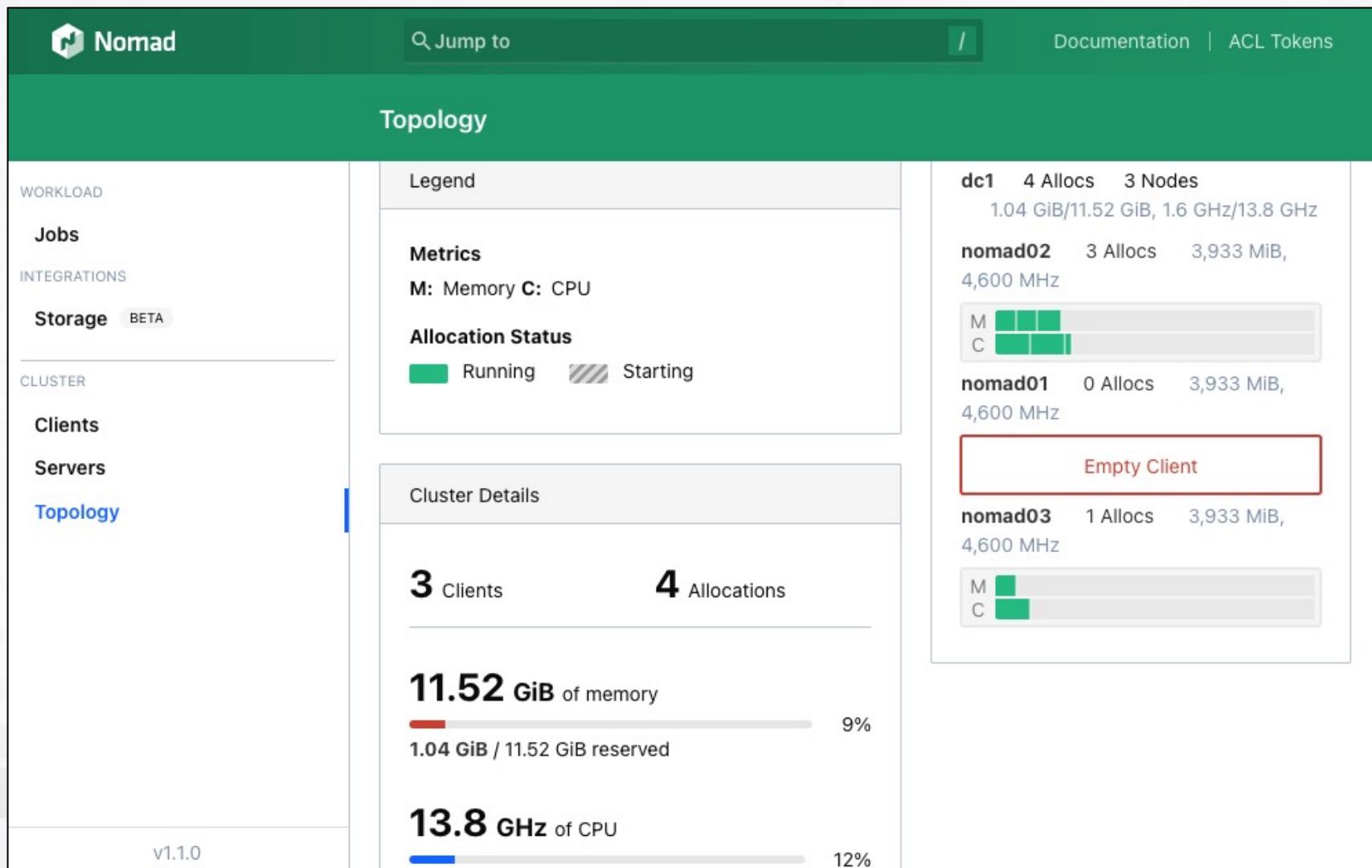
Nomad: Working principles

- Interface with Nomad via **RESTful** HTTP API.
- 5 main Nomad concepts:
 - **Job**: declarative description of tasks.
 - **Node**: servers to schedule tasks.
 - **Allocation**: tasks and nodes mapping.
 - **Deployment**: allocations trackers.
 - **Evaluation**: scheduling and status.
- Support of tokens and ACLs for **authentications**.



[3] <https://www.nomadproject.io/api-docs>

Nomad: GUI and Job example



Example_job.nomad

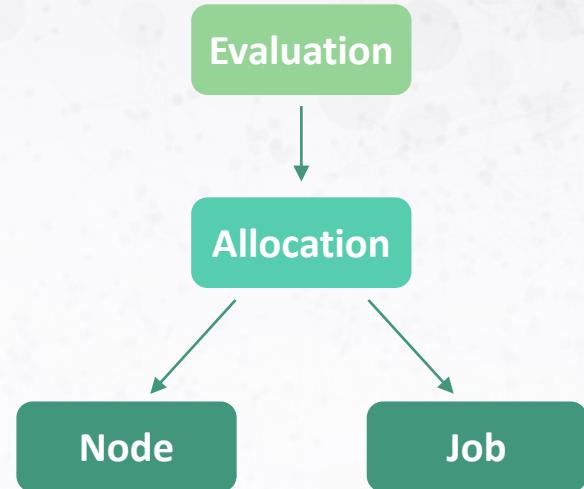
```
job "gpu-test2" {
  datacenters = ["colosseum"]
  type = "service"
  constraint {
    attribute = "${attr.unique.hostname}"
    regexp   = "dgx1"
  }
  group "smi" {
    task "smi" {
      driver = "docker"
      config {
        image = "nvidia/cuda:11.0-base"
        command = "nvidia-smi"
      }
      resources {
        device "nvidia/gpu" {
          count = 1
        }
      }
      network {
        port "web" {
          static = 81
          to = 5001
        }
      }
    }
  }
}
```

Nomad is also being considered to run and manage all Colosseum containers.

Nomad Ecosystem Implementation

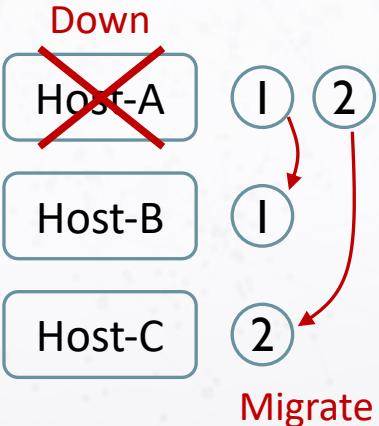
Main Entities

 HashiCorp Nomad	 HashiCorp Consul	 traefik
<ul style="list-style-type: none">• Workload orchestrator• Container manager• Simple, light, scalable	<ul style="list-style-type: none">• Service mesh tool• Service discovery/check• Secure connectivity	<ul style="list-style-type: none">• Reverse proxy• Load balancer• Completely integrated



Environment	Dev	Stage	Prod
Consul Servers	3	5	7
Nomad Servers	3	5	7
Nomad Workers (CPU)	4	4	4
Nomad Workers (GPU)	1	2	2

Servers	3	5	7
Quorum	2	3	4
Failure Tolerance	1	2	3



Possible use case examples

1. **Training** of large-scale datasets collected on Colosseum.

- **Current** data flow: Generate → Export → Train → Import → Use
- **New** data flow: Generate → Train → Use
- Much **faster** and more **efficient**.

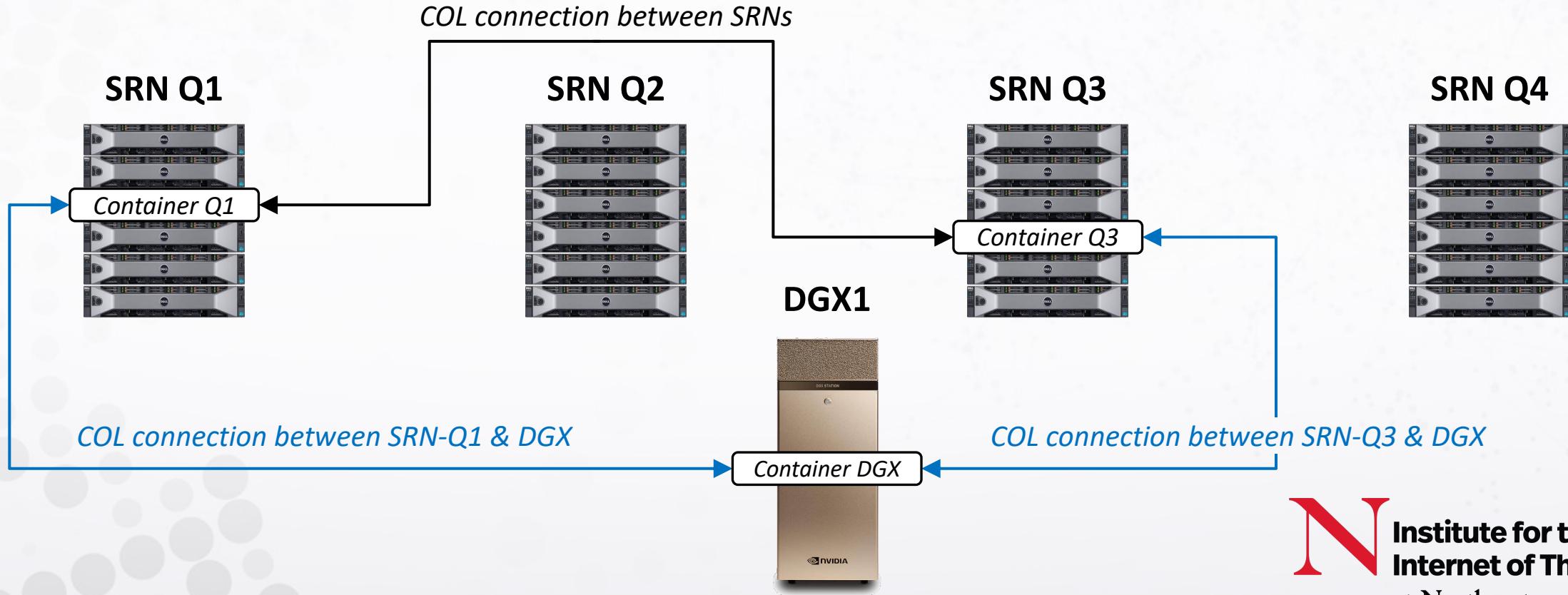
2. **Real-time** AI-driven signal processing for the full 5G protocol stack.

- **Online training** for the O-RAN xApps.
- Efficient **real-time tuning** of several users' parameters.
- Highly **scalable**.

Possible use case examples

3. Direct link between USRP and GPU to immediately process IQ samples.

- Need consideration on constrain **limits** (latency, number of users...).
- Architectural **challenges** to be solved.



Start an GPU Reservation

Home Reservations Batch Jobs Scenarios Images gladiators-admin < Reservations Request New Reservation

Experiment schedule and duration

of GPU nodes

GPU type

Docker images

Options

Reservation cost

Request New Reservation

Quad 1 1 available Quad 2 3 available Quad 3 3 available Quad 4 5 available Quad GPU 22 available

Tue 6 June 8:00 am 12:00 pm 4:00 pm 8:00 pm Wed 7 June 12:00 am 4:00 am

	Current	Cost	Remaining
Tokens	3480	24	3456

Reset Request

Check this option to reserve SRNs that can establish a link with the GPU nodes (currently SRNs available: 1).

SRN-GPU link:

Quick Start Guide FreshDesk

Demo

The screenshot shows the 'Request New Reservation' page. It includes fields for 'Name', 'Start date' (2023/06/06), 'Start time' (11:20 AM), 'Duration' (60 minutes), and a note about data transfer. It also has dropdowns for 'Number of SRNs' (0/12 max) and 'Number of GPUs' (2/8 max). Under 'Options', there's a checkbox for 'SRN-GPU link'. A summary table at the bottom shows tokens: Current 3480, Cost 24, Remaining 3456. A large blue arrow points to the 'Request' button.

View Existing Reservations

The screenshot shows the 'Reservations' section of the gladiators application. At the top, there's a banner with a message about token budget: "Your team has 3432 tokens left this week. Your team will reset to 3480 - Tue Jun 13th at 9am. Please contact Colosseum help desk in case of issues." Below this is a calendar grid for June 5-6, 2023, showing a reservation for "999: test-gpu-reservati". A red arrow points from the "Token budget for reservations" text to this message. Another red arrow points from the "Reservation list" text to the reservation entry in the table below. A third red arrow points from the "View Node status" text to the "x" button in the table header.

Token budget for reservations

Manage gladiators Reservations

Your team has 3432 tokens left this week. Your team will reset to 3480 - Tue Jun 13th at 9am
Please contact Colosseum help desk in case of issues.

+ New Reservation

Mon 5 June Tue 6 June

8:00 pm 9:00 pm 10:00 pm 11:00 pm 12:00 am 1:00 am 2:00 am 3:00 am 4:00 am 5:00 am 6:00 am 7:00 am 8:00 am 9:00 am 10:00 am 11:00 am 12:00 pm 1:00 pm 2:00 pm 3:00 pm 4:00 pm 5:00 pm 6:00 pm 7:00 pm

999: test-gpu-reservati

Export Last 7 days

Reservation Details

Type	Res ID	Name	Status	Nodes	Start	End	Duration(min)	User	⋮	x
All	999	test-gpu-reservation	Future	2 node(s)	2023/06/06 - 11:21:00 AM	2023/06/06 - 1:21:00 PM	120	gladiators-admin	⋮	x

Reservation list

View Node status

Quick Start Guide FreshDesk

View Reservation Details

The screenshot shows the 'View test-gpu-reservation' page. At the top, it displays the reservation ID (999), status (Active), number of nodes (2), and reservation time (2023/06/06 11:21:00 AM - 2023/06/06 1:21:00 PM (120 minutes)). Below this, a table lists the two nodes. Each node row contains the Node name, Image name, Hostname, Port, and Status. Red boxes highlight the Hostname, Port, and Status columns for both nodes. Red arrows point from the explanatory text below to these highlighted fields.

Node	Image	Hostname	Port	Status
GPU DGX 1	base-tensorflow:1.0	gladiators-999.service.colosseum.prod.colosseum.net	21410	Ready
GPU DGX 2				

Hostname of the Docker container with GPU nodes

Port to ssh in the container

Node status

Login example (from your pc): `ssh gladiators-999.service.colosseum.prod.colosseum.net -p 21410`

Images Management

The screenshot shows the 'Images Management' section of a web application. It includes two main sections: 'Push an image' and 'Commit an image'.

Push an image (highlighted with a red arrow pointing to the 'Push a new image' button):

- Docker Images (GPU)**
- Push a new image**
- Step 1:** Copy your docker .tar.gz image in the NAS:
`scp {path-of-your-image} gladiators-admin@file-proxy:/share/nas/gladiators/push-images/`
- Step 2:** Select the image to push:
No push images available
- Step 3:** Choose a name for the pushed image:
image-name:tag
- Step 4:** Push the new image to the registry:
Push

Commit an image (highlighted with a red arrow pointing to the 'Commit an image' button):

- Step 1:** Select the active reservation to commit:
Select the GPU reservation
- Step 2:** Choose a name for the committed image:
image-name:tag
- Step 3:** Commit the image to the registry:
Commit

At the top right, the user is identified as 'gladiators-admin'. At the bottom right, there are links for 'Quick Start Guide' and 'FreshDesk'.

Images Management

The screenshot shows the 'Images' tab selected in a navigation bar. The main content area is divided into two sections: 'Export an image' and 'Delete an image'.

Export an image:

- Step 3: Commit the image to the registry:** Contains a 'Commit' button.
- Export an image:** Sub-section with the following steps:
 - Step 1: Select the image to export:** A dropdown menu labeled 'Select the image'.
 - Step 2: Choose a name for the exported image:** An input field containing 'image-name_tag'.
 - Step 3: Export the selected image to the NAS:** A 'Export' button.
 - Step 4: Copy the docker .tar.gz image to your device:** A command line instruction: `scp gladiators-admin@file-proxy:/share/nas/gladiators/exported-images/ {path-on-your-device}`.

Delete an image:

- Delete an image:** Sub-section with the following steps:
 - Step 1: Select the image to delete:** A dropdown menu labeled 'Select the image'.
 - Step 2: Delete the selected image from the registry:** A 'Delete' button.

At the bottom, a note says: 'For further information on Docker images management, please visit [Manage Docker Containers](#)'.

GPU Links to Colosseum Resources and Documentations

- Colosseum Home Page
 - <http://colosseum.net>
 - <https://northeastern.edu/colosseum>
- Colosseum Experiment Portal
 - <https://experiments.colosseum.net>
 - <https://experiments.colosseum.net/images>
- GitHub Colosseum Sample Dockerfiles
 - <https://github.com/colosseum-wiot/colosseum-dockerfiles/>
- Freshdesk Guides
 - Logging into a GPU: [Link](#)
 - Manage Docker containers: [Link](#)
 - Connecting GPU and SRN Reservations: [Link](#)

GPU Assignment

GPU Assignment

- We are going to follow an example from Pytorch



Steps to Follow

1. Reserve a time slot for 1 DGX gpu
2. Pick the Pytorch image from common images
3. Connect to the Container using ssh

Steps to Follow

- wget the the code and the dataset on your container
- check the nvidia-smi for GPUs availability
- Run the code and have fun using python
- You can check the results by
 - VS Code / PyCharm
 - scp / rsync



Institute for the Wireless Internet of Things

at Northeastern University

Thank You! (Questions?)



Platforms for Advanced
Wireless Research



MITRE



MASSACHUSETTS
TECHNOLOGY
COLLABORATIVE



N COLOSSEUM
at Northeastern University