



Institute for the Wireless Internet of Things

at Northeastern University

GPUs Overview

Ali Saeizadeh



Platforms for Advanced
Wireless Research



MITRE



MASSACHUSETTS
TECHNOLOGY
COLLABORATIVE



N COLOSSEUM
at Northeastern University

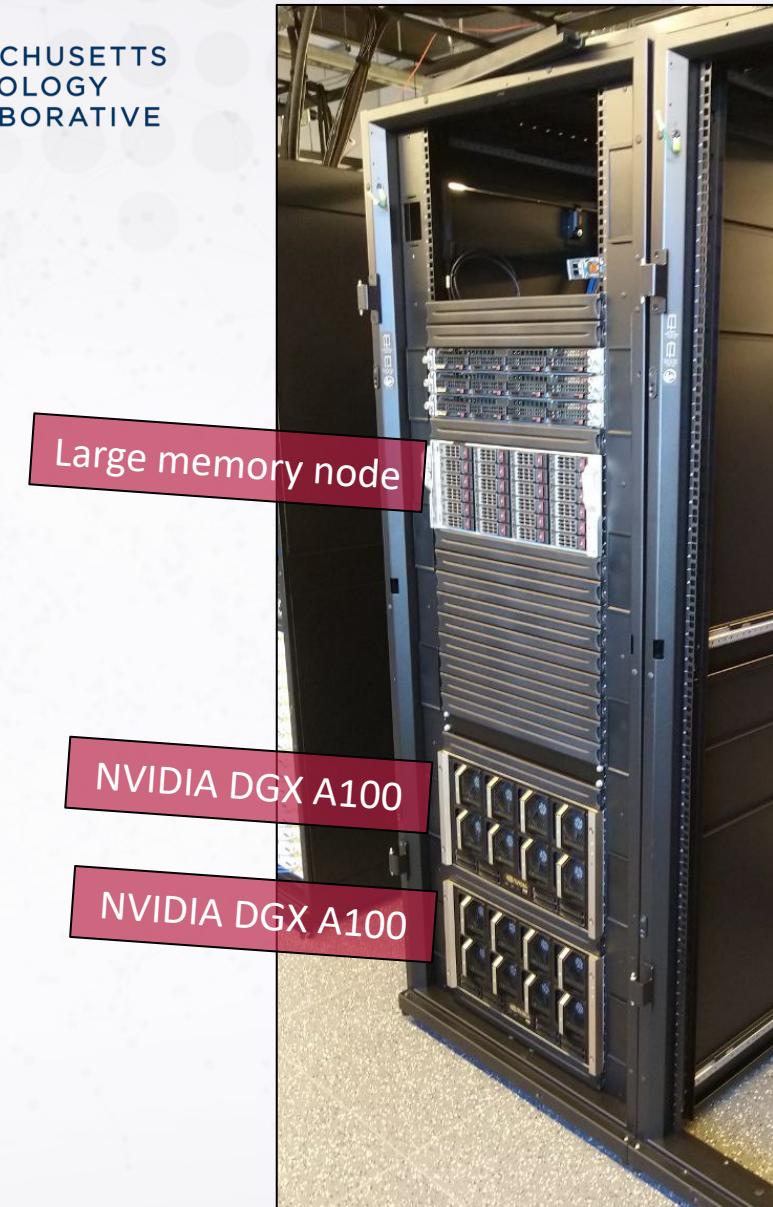
AI Jumpstart Integration



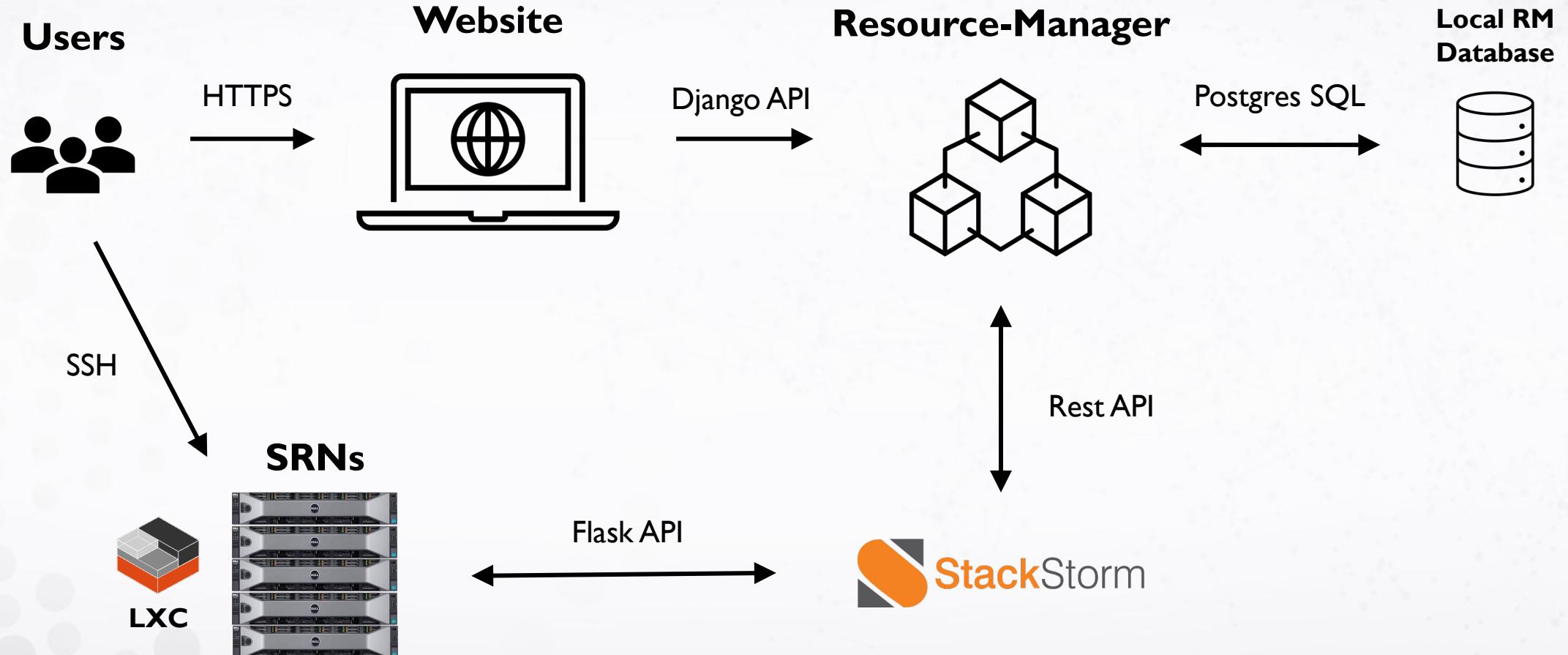
- Build GPU-accelerated software-defined, cloud-native applications for the 5G vRAN
 - 2x NVIDIA DGX A100 (8 GPUs each, 40GB RAM)
 - 1x Large memory node (6 GPUs, 3TB of RAM)
 - 1x MLX InfiniBand switch



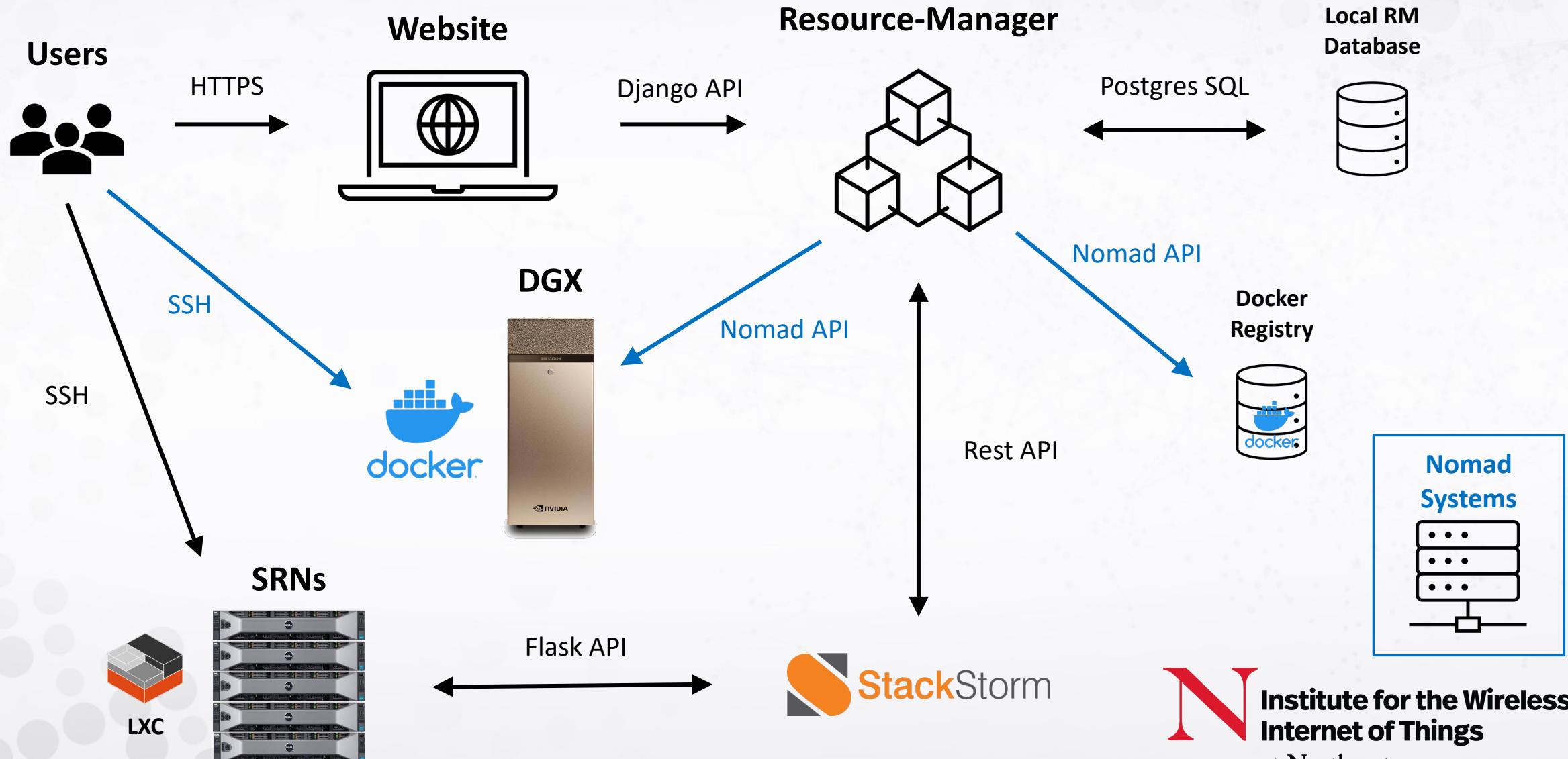
How do we **integrate** the new rack into Colosseum?



Current SRN reservation flow diagram



New GPU Reservation Flow Diagram



Nomad: Overview

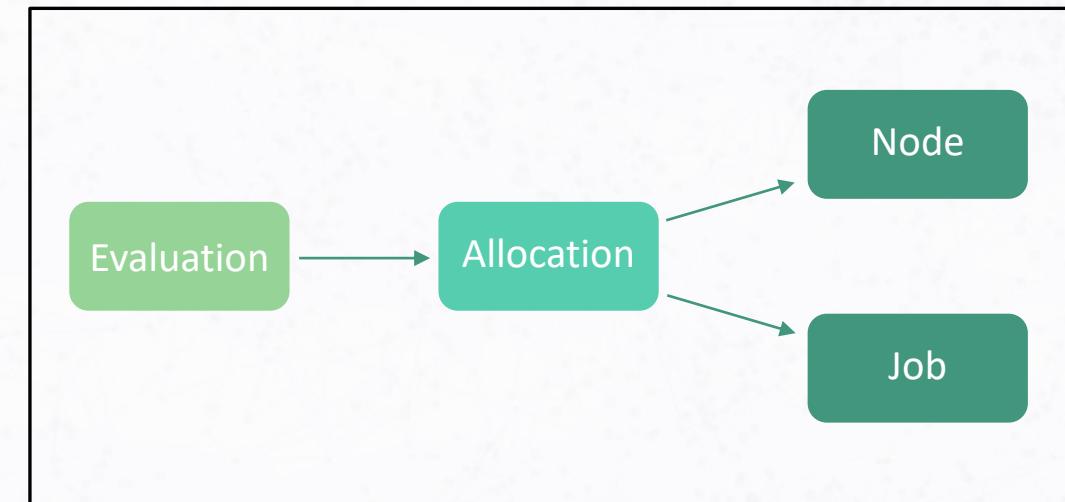


- Workload **orchestrator** to deploy and manage containers.
- Built-in support for **GPU** workloads and device plugins.
- **HashiCorp** ecosystem (Vault, Consul...).
- Compare to the more popular  **kubernetes**
 - Smaller scope
 - More simple and lightweight
 - More scalable (5k vs 10k nodes)
 - Flexible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



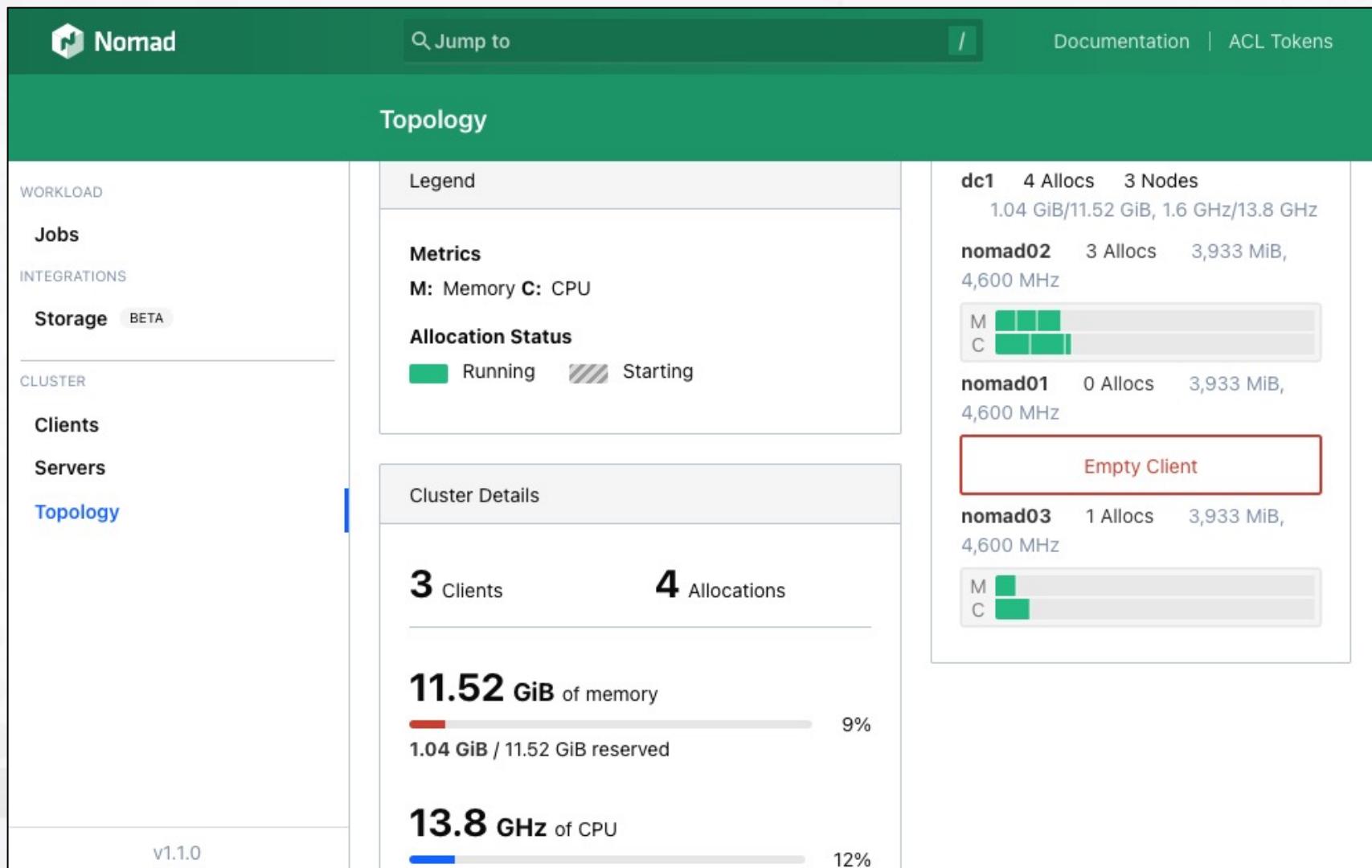
Nomad: Working principles

- 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**.



More info: <https://www.nomadproject.io/api-docs>

Nomad: GUI and Job example



Example_job.nomad

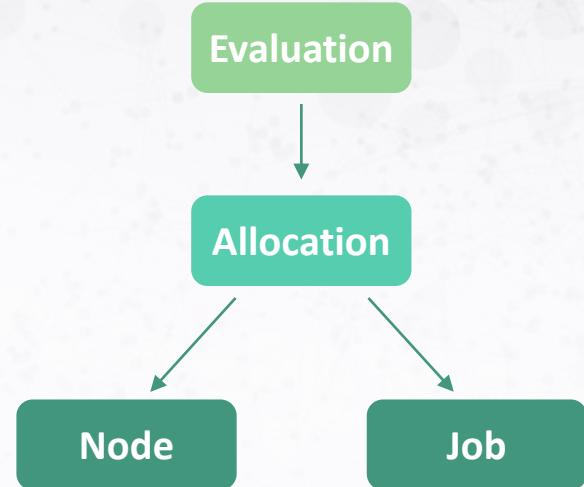
```
job "gpu-test2" {
  datacenters = ["colosseum"]
  type = "service"
  constraint {
    attribute = "${attr.unique.hostname}"
    regexp   = "dgx1"
  }
  group "smi" {
    task "smi" {
      driver = "docker"
      config {
        image = "nvidia/cuda:11.0-base"
        command = "nvidia-smi"
      }
      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

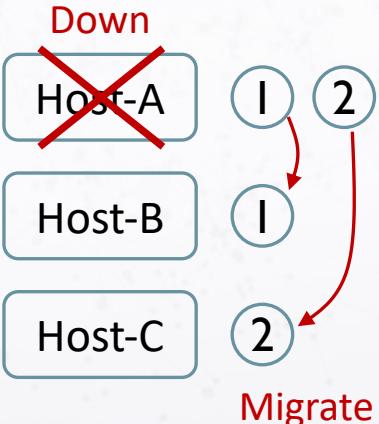
Main Entities

 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**.

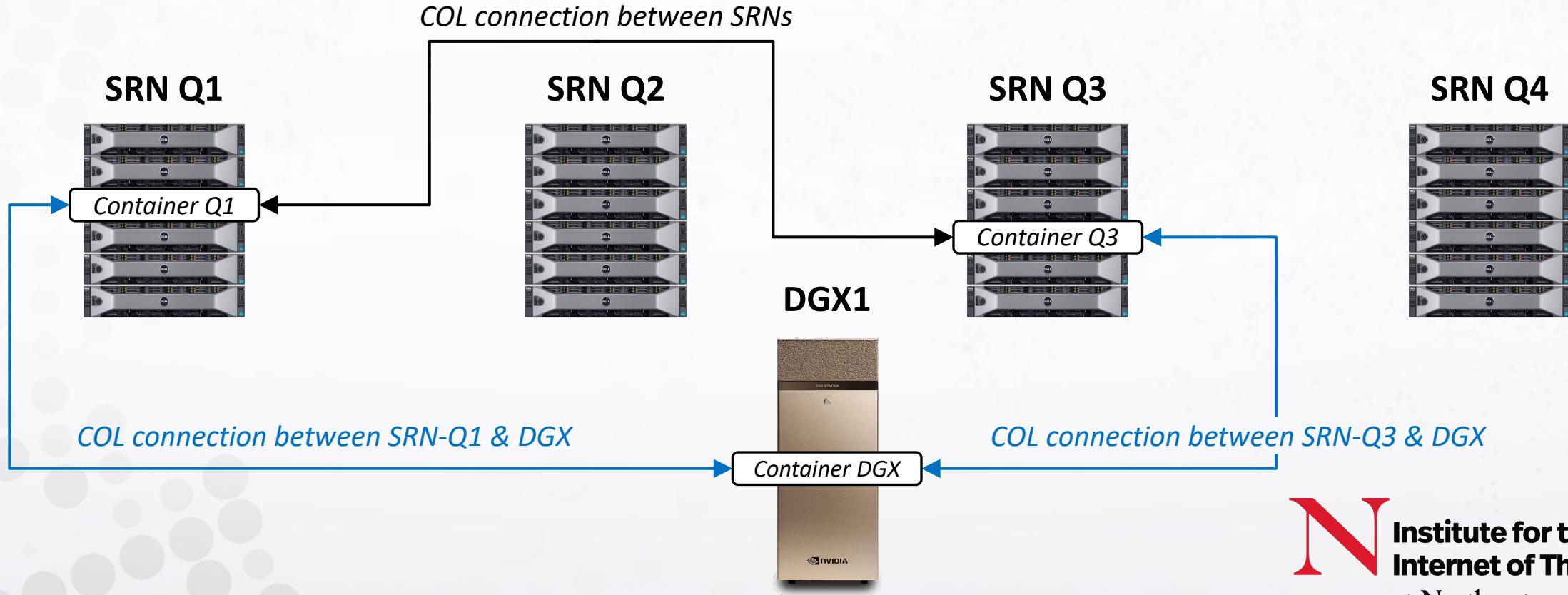
2. **Real-time** AI-driven signal processing for the full 5G protocol stack.

- **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 a GPU Reservation

Home Reservations Batch Jobs Scenarios Images Blueprints gladiators-admin

< Reservations

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 'GPU node type', 'DGX' is selected. The 'Default GPU image' is set to 'base-pytorch:1.0'. A checkbox for 'SRN-GPU link' is present with a note about SRN availability. On the right, a resource grid shows availability for Quads 1-4 and Quad GPU across two days. A summary table at the bottom shows tokens: Current 3480, Cost 24, Remaining 3456. Buttons for 'Reset' and 'Request' are at the bottom right. Red arrows point from the text labels to their corresponding form fields or sections.

View Existing Reservations

The screenshot shows the 'Manage gladiators Reservations' interface. At the top, a banner displays the 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.' A red arrow points from the text 'Token budget for reservations' to this banner. Below the banner is a timeline grid for 'Mon 5 June' and 'Tue 6 June'. A blue box highlights the entry '999: test-gpu-reservati' on Tuesday. A red box highlights the 'Reservation Details' table below, which lists the reservation '999' for 'test-gpu-reservation' with a duration of 120 minutes. Red arrows point from the text 'Reservation list' to the table header and from 'View Node status' to the delete button in the table row. The bottom navigation bar includes links for 'Quick Start Guide' and 'FreshDesk'.

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
Interactive	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]

Quick Start Guide FreshDesk

View Reservation Details

The screenshot shows the Colosseum interface with the following details:

- ID:** 1499
- Status:** Active
- Nodes:** 2
- Time:** 2024/03/21 12:16:00 PM - 2024/03/21 5:16:00 PM (300 minutes)
- Node:** SRN-11
- Image:** webinar-interactive-v1
- Hostname:** wineslab-011
- Ports of the Docker container:**

Name	External	Internal
ssh	30662	22
dp1	23513	5201
dp2	25840	5202
dp3	20337	5203
- Status:** Allocated
- GPU DGX 1 base-ssh:1.0**
- Copy SSH cmd to clipboard** button (highlighted with a red box and arrow)
- wineslab-1499.service.colosseum.prod.colosseum.net** (highlighted with a red box and arrow)
- Ready** button (highlighted with a red box and arrow)
- Node status** (highlighted with a red box and arrow)

At the bottom, there are links for [Quick Start Guide](#) and [FreshDesk](#).

Login example (from your pc): `ssh wineslab-1499.service.colosseum.prod.colosseum.net -p 30662`

2 types of ports:

- **ssh**: to ssh into the container.
- **dp** (dynamic port): to interact with external apps from SRNs (e.g., iPerf server).

Images Management

The screenshot shows the 'Images Management' page with 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' section):

- 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' section):

- 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

Top navigation bar: Home, Reservations, Batch Jobs, Scenarios, **Images**, Blueprints. Top right: gladiators-admin, Logout.

Images Management

The screenshot shows a web-based application interface for managing Docker images. At the top, there is a navigation bar with links: Home, Reservations, Batch Jobs, Scenarios, Images (which is highlighted in green), and Blueprints. On the right side of the header, it says "gladiators-admin" and has a user icon.

Export an image: A red arrow points to the "Export an image" section. It contains four steps:

- Step 3: Commit the image to the registry: [Commit](#)
- Step 1: Select the image to export: [Select the image](#)
- Step 2: Choose a name for the exported image:
- Step 3: Export the selected image to the NAS: [Export](#)

Delete an image: A red arrow points to the "Delete an image" section. It also contains four steps:

- Step 1: Select the image to delete: [Select the image](#)
- Step 2: Delete the selected image from the registry: [Delete](#)
- Step 3: Commit the image to the registry: [Commit](#)
- Step 4: Copy the docker .tar.gz image to your device:
`scp gladiators-admin@file-proxy:/share/nas/gladiators/exported-images/ {path-on-your-device}`

At the bottom of the page, there is a note: "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