



[www.chameleoncloud.org](http://www.chameleoncloud.org)

## CHAMELEON: CREATING AN ECOSYSTEM FOR EXPERIMENTAL COMPUTER SCIENCE

**Kate Keahey**

Mathematics and CS Division, Argonne National Laboratory  
CASE, University of Chicago  
[keahey@anl.gov](mailto:keahey@anl.gov)

*November 11, 2018*  
*INDIS Workshop*

NOVEMBER 14, 2018

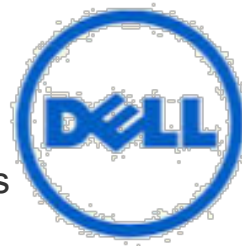
I



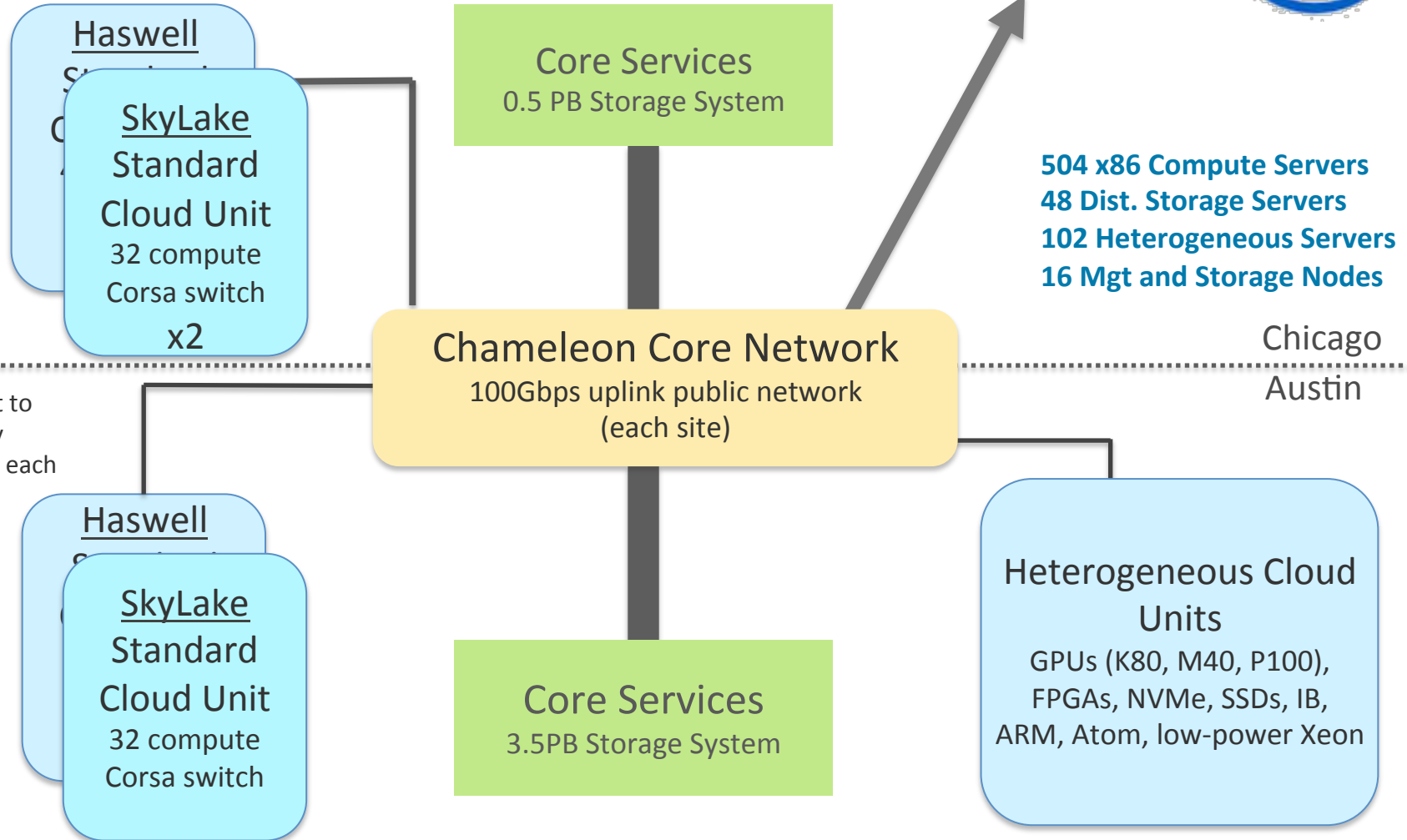
# CHAMELEON IN A NUTSHELL

- ▶ **Deeply reconfigurable:** “As close as possible to having it in your lab”
  - ▶ Deep reconfigurability (bare metal) and isolation
  - ▶ Power on/off, reboot from custom kernel, serial console access, etc.
  - ▶ But also – modest KVM cloud for ease of use
- ▶ **Combining large-scale and diversity:** “Big Data, Big Compute research”
  - ▶ **Large-scale:** ~large homogenous partition (~15,000 cores), 5 PB of storage distributed over 2 sites connected with 100G network...
  - ▶ ...and **diverse:** ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
  - ▶ **Coming soon:** more storage, more accelerators
- ▶ Blueprint for a **sustainable** production testbed: “cost-effective to deploy, operate, and enhance”
  - ▶ Powered by OpenStack with bare metal reconfiguration (Ironic)
  - ▶ Chameleon team contribution recognized as official OpenStack component
- ▶ **Open, collaborative, production** testbed for **Computer Science Research**
  - ▶ Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
  - ▶ Currently 2,700+ users, 450+ projects, 100+ institutions

# CHAMELEON HARDWARE



To GENI and other partners



# CHAMELEON HARDWARE (DETAILS)

- ▶ “Start with large-scale homogenous partition”
  - ▶ 12 Haswell Standard Cloud Units (48 node racks), each with 42 Dell R630 compute servers with dual-socket Intel Haswell processors (24 cores) and 128GB RAM and 4 Dell FX2 storage servers with 16 2TB drives each; Force10 s6000 OpenFlow-enabled switches 10Gb to hosts, 40Gb uplinks to Chameleon core network
  - ▶ 2 SkyLake Standard Cloud Units (32 node racks); Corsa (DP2400 & DP2200) switches, 100Gb uplinks to Chameleon core network
  - ▶ Allocations can be an entire rack, multiple racks, nodes within a single rack or across racks (e.g., storage servers across racks forming a Hadoop cluster)
- ▶ Shared infrastructure
  - ▶ 3.6 + 0.5 PB global storage, 100Gb Internet connection between sites
- ▶ “Graft on heterogeneous features”
  - ▶ Infiniband with SR-IOV support, High-mem, NVMe, SSDs, GPUs (22 nodes), FPGAs (4 nodes)
  - ▶ ARM microservers (24) and Atom microservers (8), low-power Xeons (8)
- ▶ Coming soon: more nodes (CascadeLake), and more accelerators



# REQUIREMENTS FOR EXPERIMENTAL WORKFLOW

discover  
resources

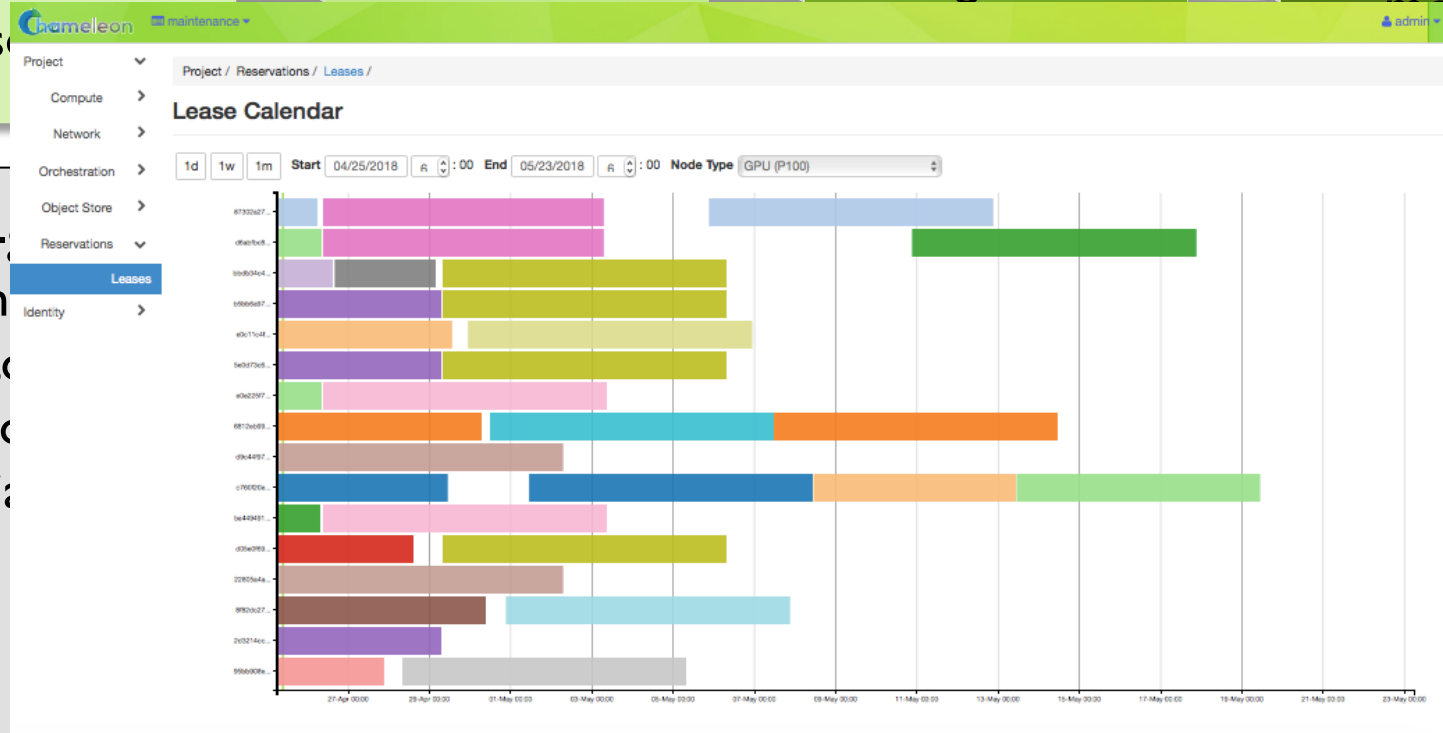
allocate

configure and

monitor

- Fine-grained
- Composable
- Up-to-date
- Versioned
- Verifiable

ware  
grained  
ation  
gate and



Isolation

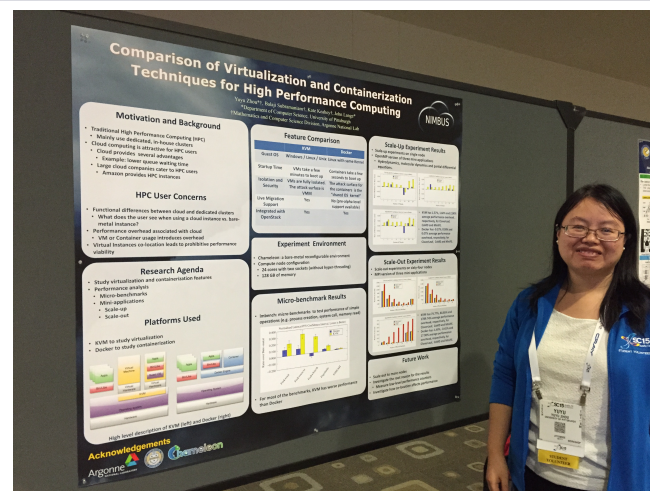
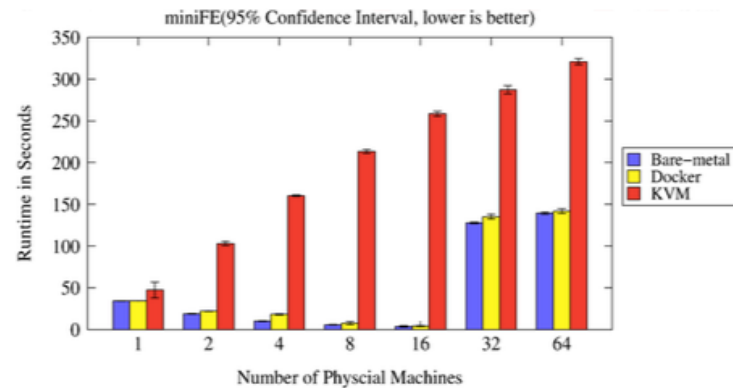
$$\text{CHI} = 65\% * \text{OpenStack} + 10\% * \text{G5K} + 25\% * \text{''special sauce''}$$

# NEWEST CAPABILITIES

- ▶ Networking:
  - ▶ **Multi-tenant networking** allows users to provision isolated L2 VLANs and manage their own IP address space (since Fall 2017)
  - ▶ **Stitching** dynamic VLANs from Chameleon to external partners (ExoGENI, ScienceDMZs) (since Fall 2017)
  - ▶ VLANs + AL2S connection between UC and TACC for **100G experiments** (since Spring 2018)
  - ▶ **BYOC— Bring Your Own Controller**: isolated user controlled virtual OpenFlow switches (since Summer 2018)
- ▶ And many others: new lease management features, multi-region configuration, power consumption metrics, whole disk image boot for ARM nodes, serial console access, appliances, upgrades, usability improvements, etc.

# VIRTUALIZATION OR CONTAINERIZATION?

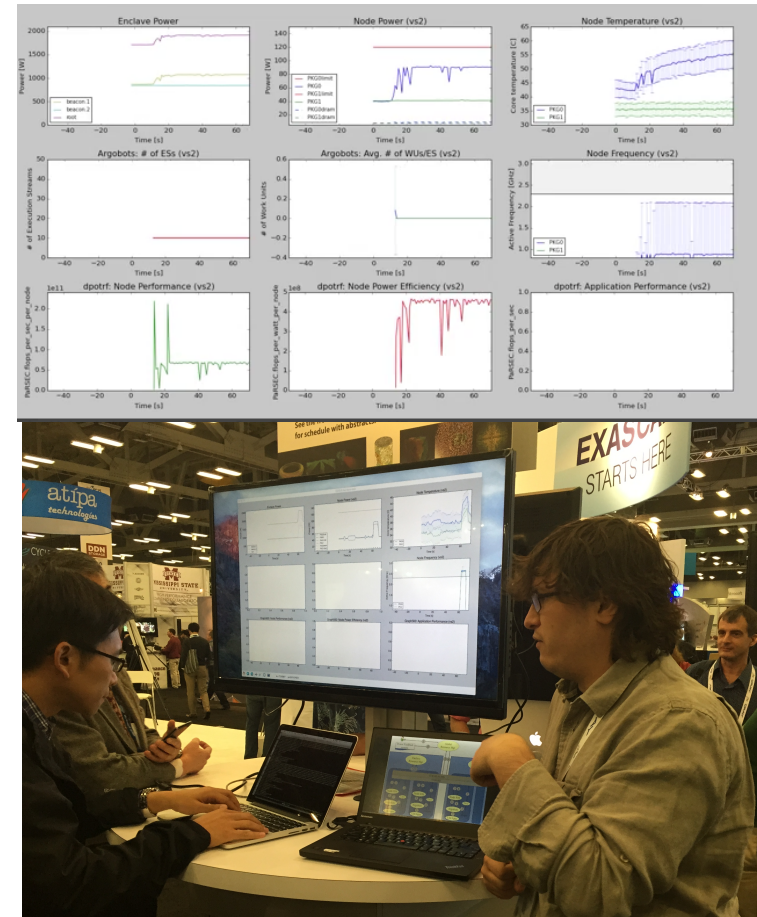
- ▶ Yuyu Zhou, University of Pittsburgh
- ▶ Research: lightweight virtualization
- ▶ Testbed requirements:
  - ▶ Bare metal reconfiguration, isolation, and serial console access
  - ▶ The ability to “save your work”
  - ▶ Support for large scale experiments
  - ▶ Up-to-date hardware



SC15 Poster: “Comparison of Virtualization and Containerization Techniques for HPC”

# EXASCALE OPERATING SYSTEMS

- ▶ Swann Perarnau, ANL
- ▶ Research: exascale operating systems
- ▶ Testbed requirements:
  - ▶ Bare metal reconfiguration
  - ▶ Boot from custom kernel with different kernel parameters
  - ▶ Fast reconfiguration, many different images, kernels, params
  - ▶ Hardware: accurate information and control over changes, performance counters, many cores
  - ▶ Access to same infrastructure for multiple collaborators



*HPPAC'16 paper: “Systemwide Power Management with Argo”*

# CLASSIFYING CYBERSECURITY ATTACKS

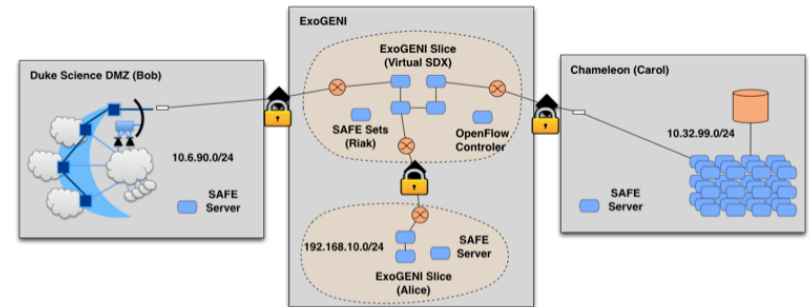
- ▶ Jessie Walker & team, University of Arkansas at Pine Bluff (UAPB)
- ▶ Research: modeling and visualizing multi-stage intrusion attacks (MAS)
- ▶ Testbed requirements:
  - ▶ Easy to use OpenStack installation
  - ▶ A selection of pre-configured images
  - ▶ Access to the same infrastructure for multiple collaborators





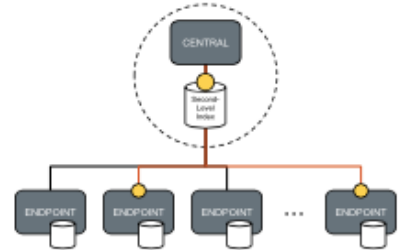
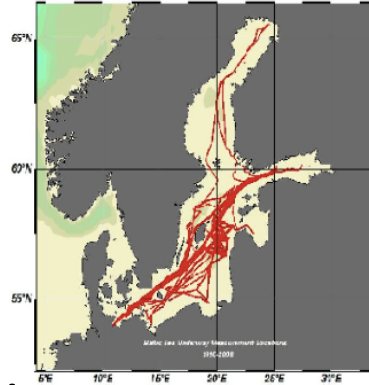
# CREATING DYNAMIC SUPERFACILITIES

- ▶ NSF CICI SAFE, Paul Ruth, RENCI-UNC Chapel Hill
- ▶ Creating trusted facilities
  - ▶ Automating trusted facility creation
  - ▶ Virtual Software Defined Exchange (SDX)
  - ▶ Secure Authorization for Federated Environments (SAFE)
- ▶ Testbed requirements
  - ▶ Creation of dynamic VLANs and wide-area circuits
  - ▶ Support for slices and network stitching
  - ▶ Managing complex deployments



# DATA SCIENCE RESEARCH

- ▶ ACM Student Research Competition semi-finalists:
  - ▶ Blue Keleher, University of Maryland
  - ▶ Emily Herron, Mercer University
- ▶ Searching and image extraction in research repositories
- ▶ Testbed requirements:
  - ▶ Access to distributed storage in various configurations
  - ▶ State of the art GPUs
  - ▶ Easy to use appliances and complex deployments



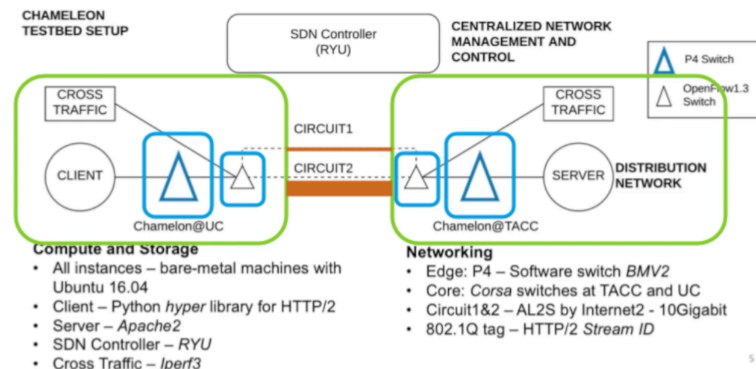
Our Method: hierarchical hybrid  
featuring "collapsed" second-  
level index (SLI)

- SLI references endpoints, not docs, and contains a summary subset of terms
- + Some storage burden on endpoints, but still very low per endpoint
- + Lower storage burden on central servers



# ADAPTIVE BITRATE VIDEO STREAMING

- ▶ Divyashri Bhat, UMass Amherst
- ▶ Research: application header based traffic engineering using P4
- ▶ Testbed requirements:
  - ▶ Distributed testbed facility
  - ▶ BYOC – the ability to write an SDN controller specific to the experiment
  - ▶ Multiple connections between distributed sites
- ▶ <https://vimeo.com/297210055>



LCN'18: “Application-based QoS support with P4 and OpenFlow”



# BUILDING AN ECOSYSTEM

- ▶ Helping hardware providers interact
  - ▶ Bring Your Own Hardware (BYOH)
  - ▶ CHI-in-a-Box: deploy your own Chameleon site
- ▶ Helping scientists interact
  - ▶ Leveraging the common denominator
  - ▶ Integrating tools for experiment management
  - ▶ Making reproducibility easier
  - ▶ Facilitating sharing

# CHI-IN-A-BOX

- ▶ CHI-in-a-box: packaging a commodity-based testbed
- ▶ CHI-in-a-box scenarios
  - ▶ **Testbed extension:** join the Chameleon testbed: generalize and package + define operations models
  - ▶ **Part-time extension:** define and implement contribution models
  - ▶ **New testbed:** generalize policies
- ▶ Understanding the support cost model
- ▶ Available since Summer 2018
- ▶ **New Associate Site at Northwestern!**
  - ▶ Nodes with 100G network cards



# REPRODUCIBILITY DILEMMA

*Should I invest in making my experiments repeatable?*



*Should I invest in more new research instead?*

- ▶ **Reproducibility as side-effect:** lowering the cost of repeatable research
  - ▶ Example: Linux “history” command
  - ▶ From a meandering scientific process to a recipe
- ▶ **Reproducibility by default:** documenting the process via interactive papers

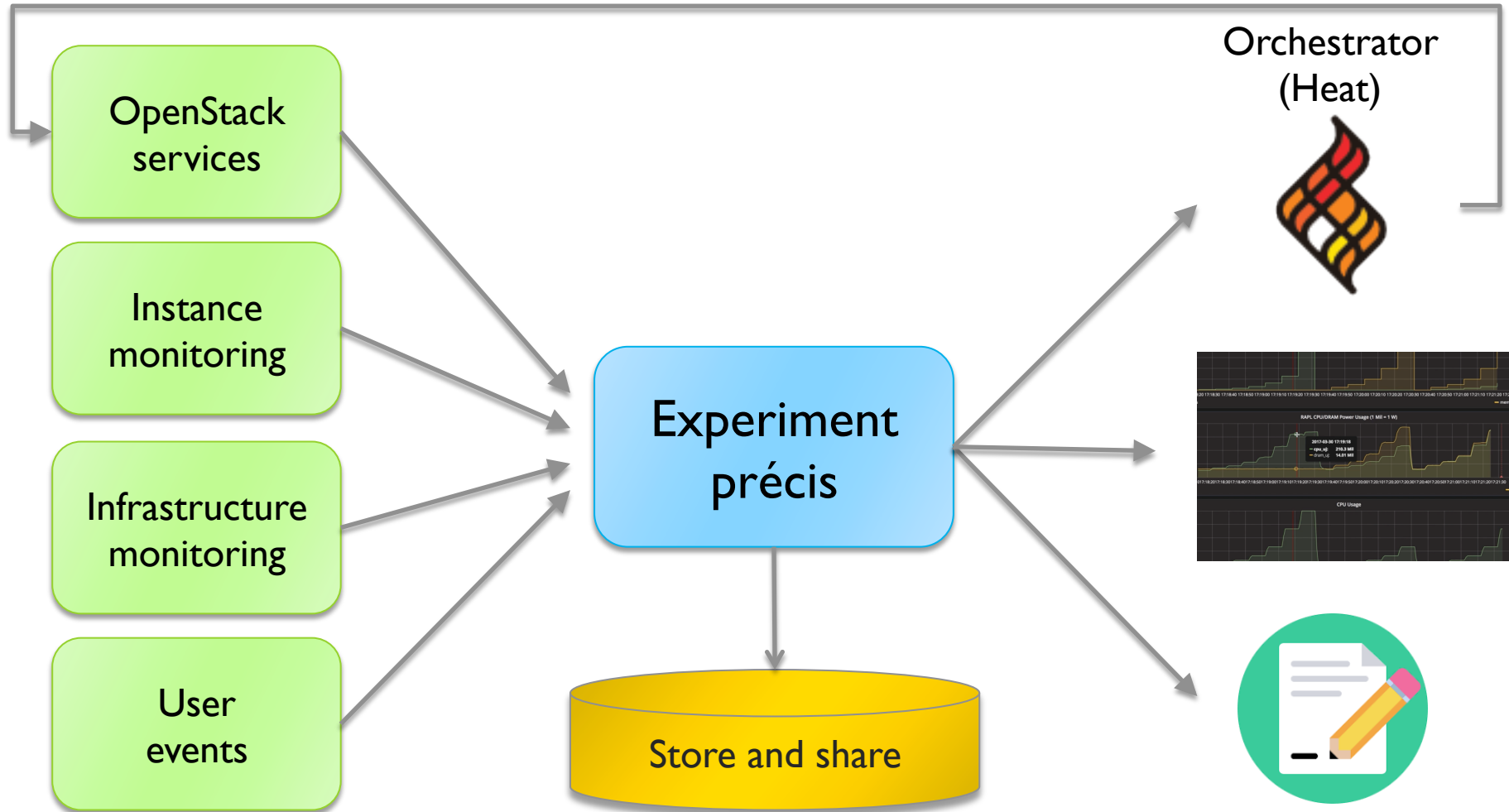
# REPEATABILITY MECHANISMS IN CHAMELEON

- ▶ Testbed versioning (collaboration with Grid'5000)
  - ▶ Based on representations and tools developed by G5K
  - ▶ >50 versions since public availability – and counting
  - ▶ Still working on: better firmware version management
- ▶ Appliance management
  - ▶ Configuration, versioning, publication
  - ▶ Appliance meta-data via the appliance catalog
  - ▶ Orchestration via OpenStack Heat
- ▶ Monitoring and logging
- ▶ **However... the user still has to keep track of this information**

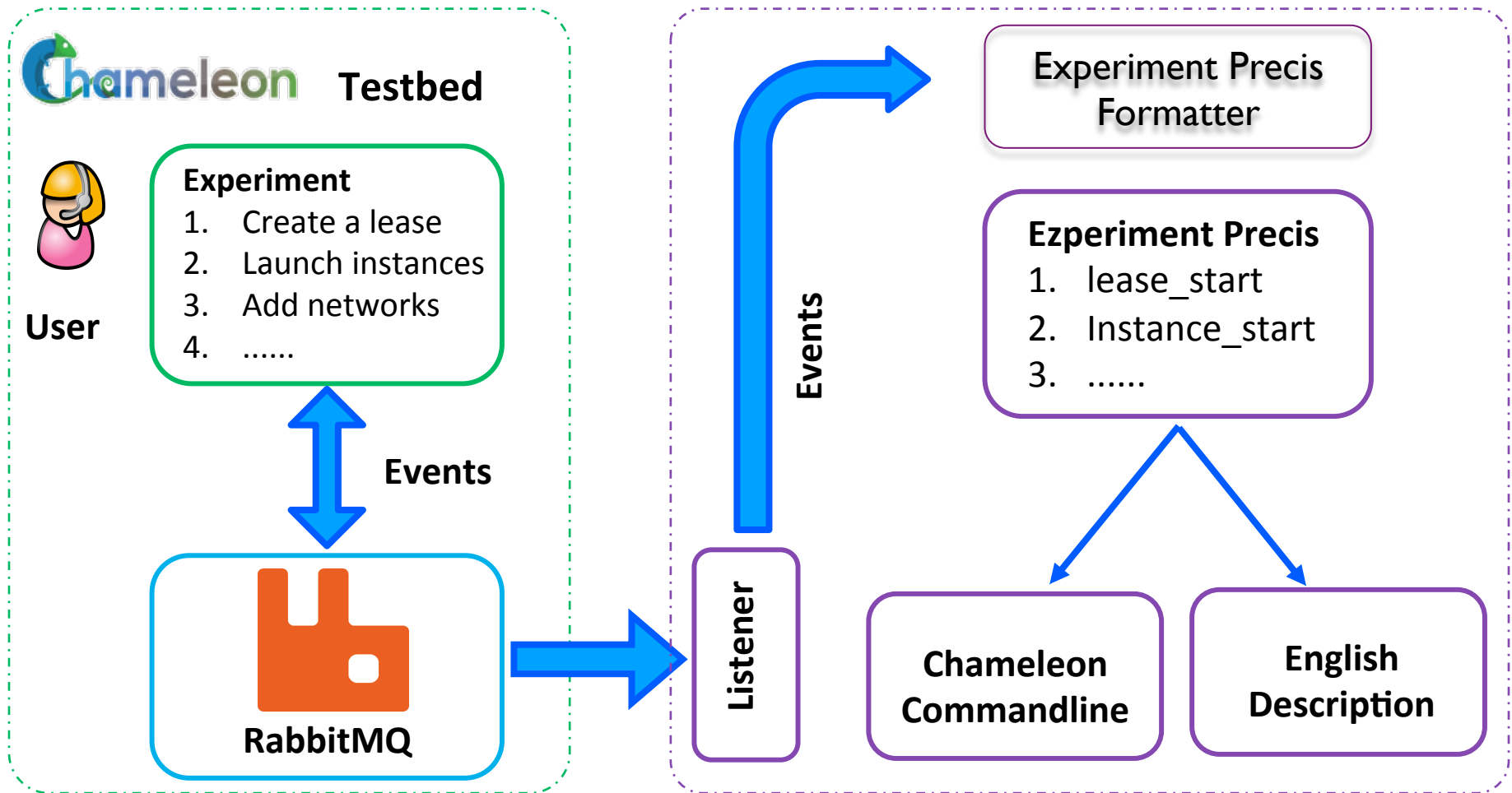
# KEEPING TRACK OF EXPERIMENTS

- ▶ Everything in a testbed is a recorded event
    - ▶ The resources you used
    - ▶ The appliance/image you deployed
    - ▶ The monitoring information your experiment generated
    - ▶ Plus any information you choose to share with us: e.g., “start power\_exp\_23” and “stop power\_exp\_23”
- 
- ▶ **Experiment précis:** information about your experiment made available in a “consumable” form

# REPEATABILITY: EXPERIMENT PRÉCIS

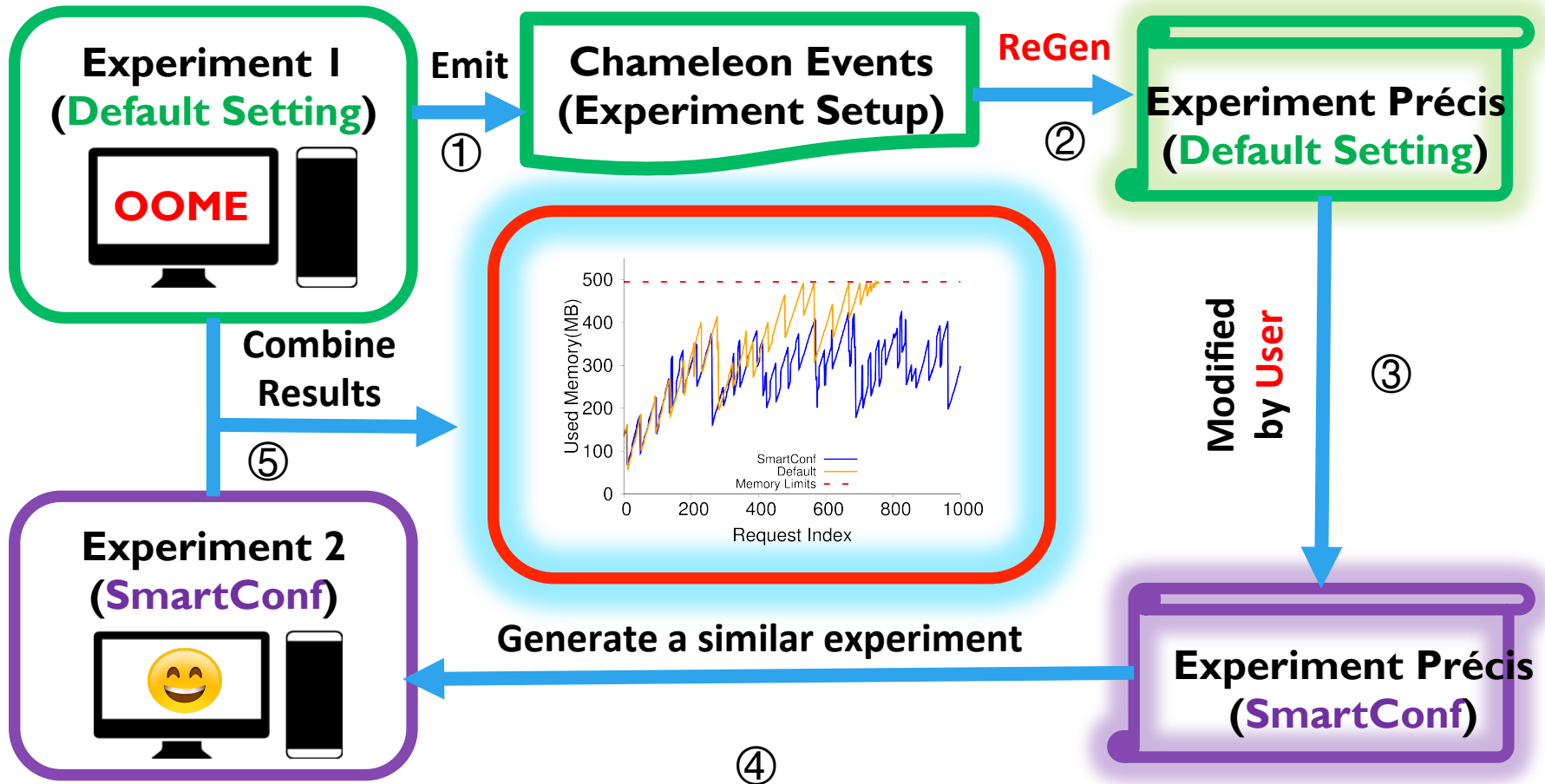


# EXPERIMENT PRÉCIS IMPLEMENTATION



*Come see our SCI8 poster: "Reproducibility as Side-Effect"*

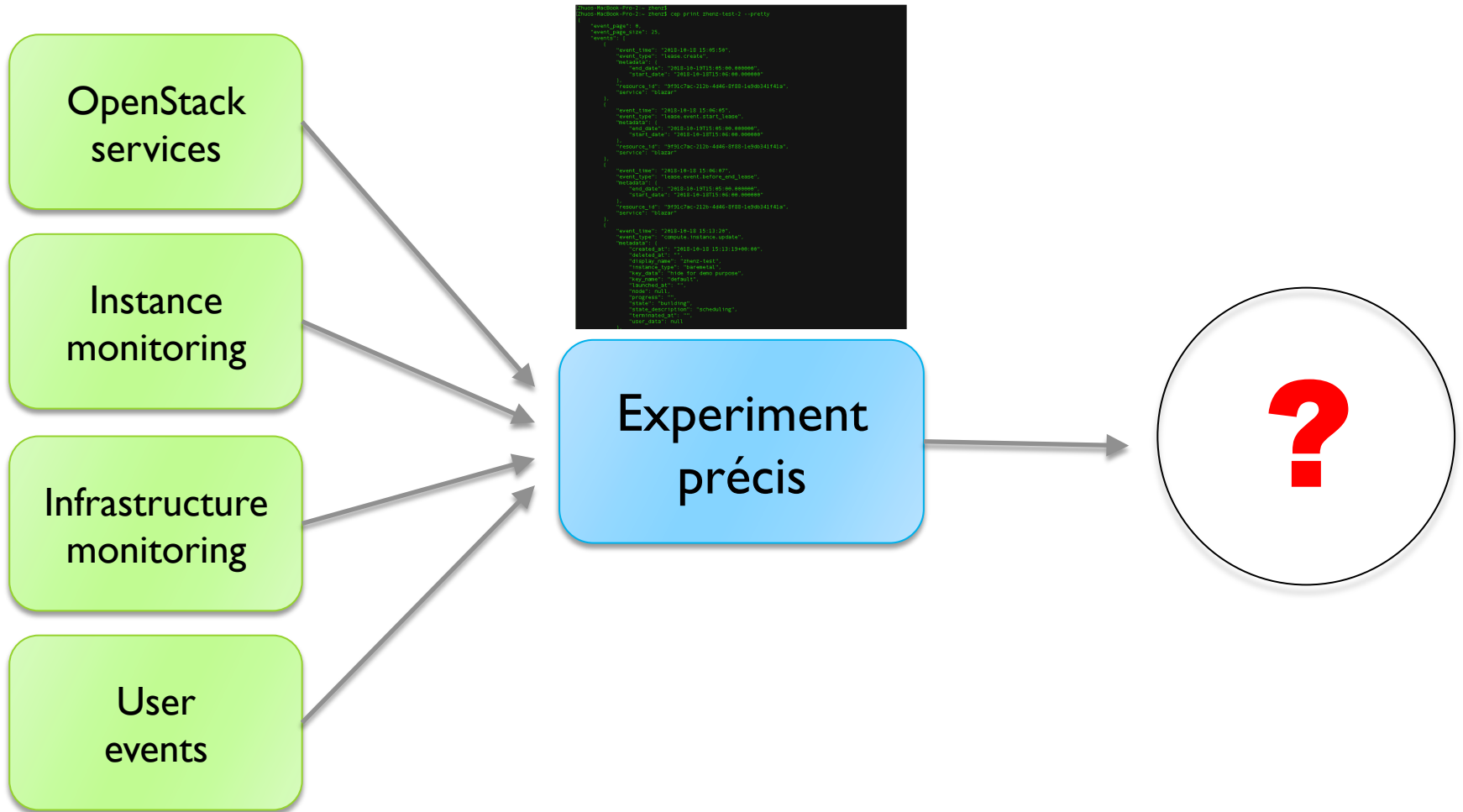
# EXPERIMENT PRÉCIS: A CASE STUDY



Based on Wang et al., Understanding and Auto-Adjusting Performance-Sensitive Configurations. ASPLOS, 2018



## REPEATABILITY: EXPERIMENT PRÉCIS



# ACTIVE PAPERS: WHAT DOES IT MEAN TO DOCUMENT A PROCESS?

## ► Requirements

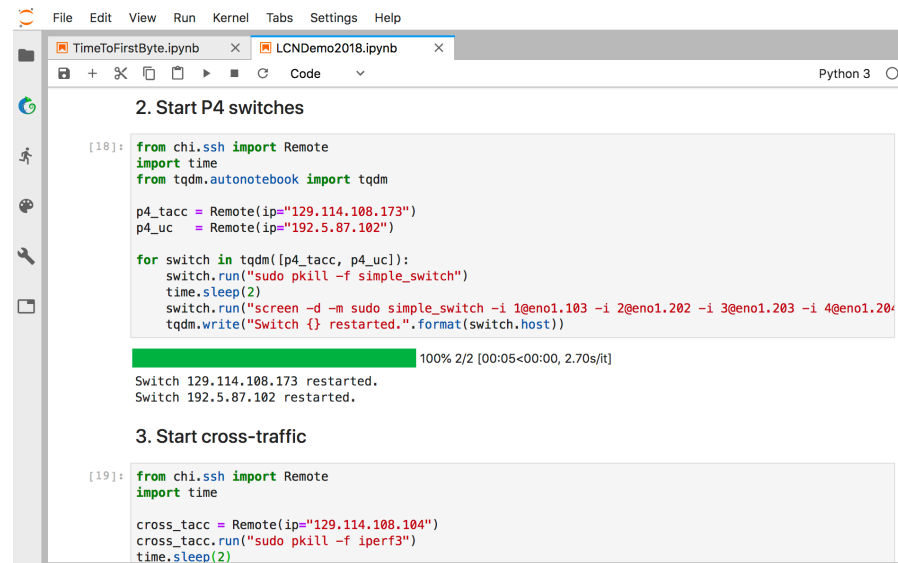
- Easy to work with: human readable/modifiable format
- Integrates well with ALL aspects of experiment management
- Bit by bit replay – allows for bit by bit modification (and introspection) as well – element of interactivity
- Support story telling: allows you to explain your experiment design and methodology choices
- Has a direct relationship to the actual paper that gets written
- Can be version controlled
- Sustainable, a popular open source choice

## ► Implementation options

- Orchestrators: Heat, the dashboard, and OpenStack Flame
- Notebooks: Jupyter, Nextjournal

# COMBINING THE EASE OF NOTEBOOKS AND THE POWER OF A SHARED PLATFORM

- ▶ Combining Jupyter with Chameleon
  - ▶ Storytelling with Jupyter: ideas/text, process/code, results
  - ▶ Chameleon shared experimental platform
- ▶ Chameleon/Jupyter integration
  - ▶ Alternative interface
  - ▶ All the main testbed functions
  - ▶ “Hello World” template
  - ▶ Save&share via object store
- ▶ Jupyter.chamelecloud.org
- ▶ Screencast of a complex experiment
  - ▶ <https://vimeo.com/297210055>



The screenshot shows a Jupyter Notebook window with two tabs: 'TimeToFirstByte.ipynb' and 'LCNDemo2018.ipynb'. The active tab is 'LCNDemo2018.ipynb', which displays Python code for starting P4 switches and cross-traffic. The code is organized into sections: '2. Start P4 switches' and '3. Start cross-traffic'. The '2. Start P4 switches' section includes code to connect to two remote hosts (129.114.108.173 and 192.5.87.102) and start a switch on each. The '3. Start cross-traffic' section includes code to connect to a third remote host (129.114.108.104) and start cross-traffic. The notebook interface shows the code being executed, with a progress bar indicating 100% completion for the first section. The output of the first section shows that the switches were restarted successfully.

```
[18]: from chi.ssh import Remote
import time
from tqdm.autonotebook import tqdm

p4_tacc = Remote(ip="129.114.108.173")
p4_uc = Remote(ip="192.5.87.102")

for switch in tqdm([p4_tacc, p4_uc]):
    switch.run("sudo pkill -f simple_switch")
    time.sleep(2)
    switch.run("screen -d -m sudo simple_switch -i 1@eno1.103 -i 2@eno1.202 -i 3@eno1.203 -i 4@eno1.204")
    tqdm.write("Switch {} restarted.".format(switch.host))

100% 2/2 [00:05<00:00, 2.70s/it]

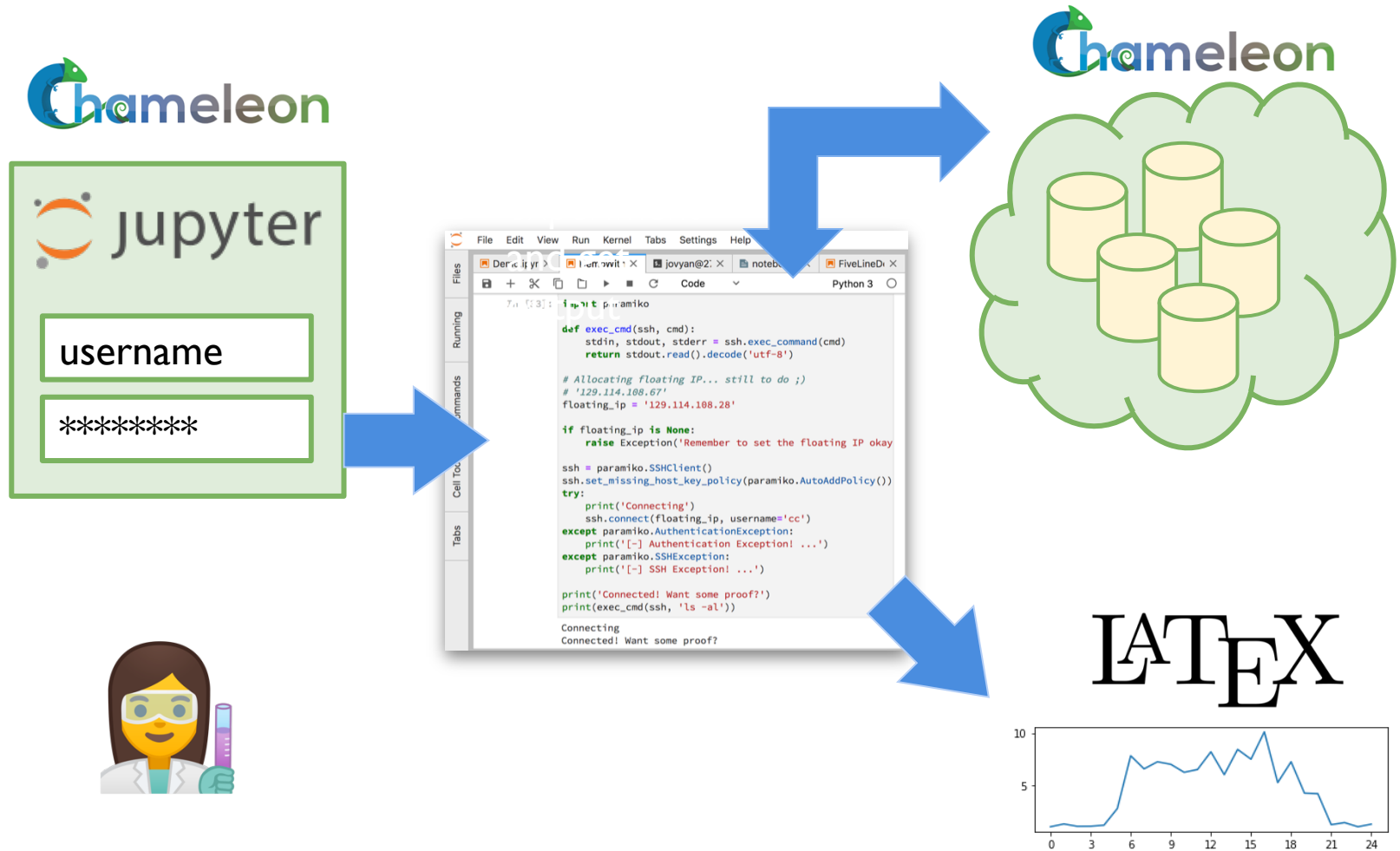
Switch 129.114.108.173 restarted.
Switch 192.5.87.102 restarted.

3. Start cross-traffic

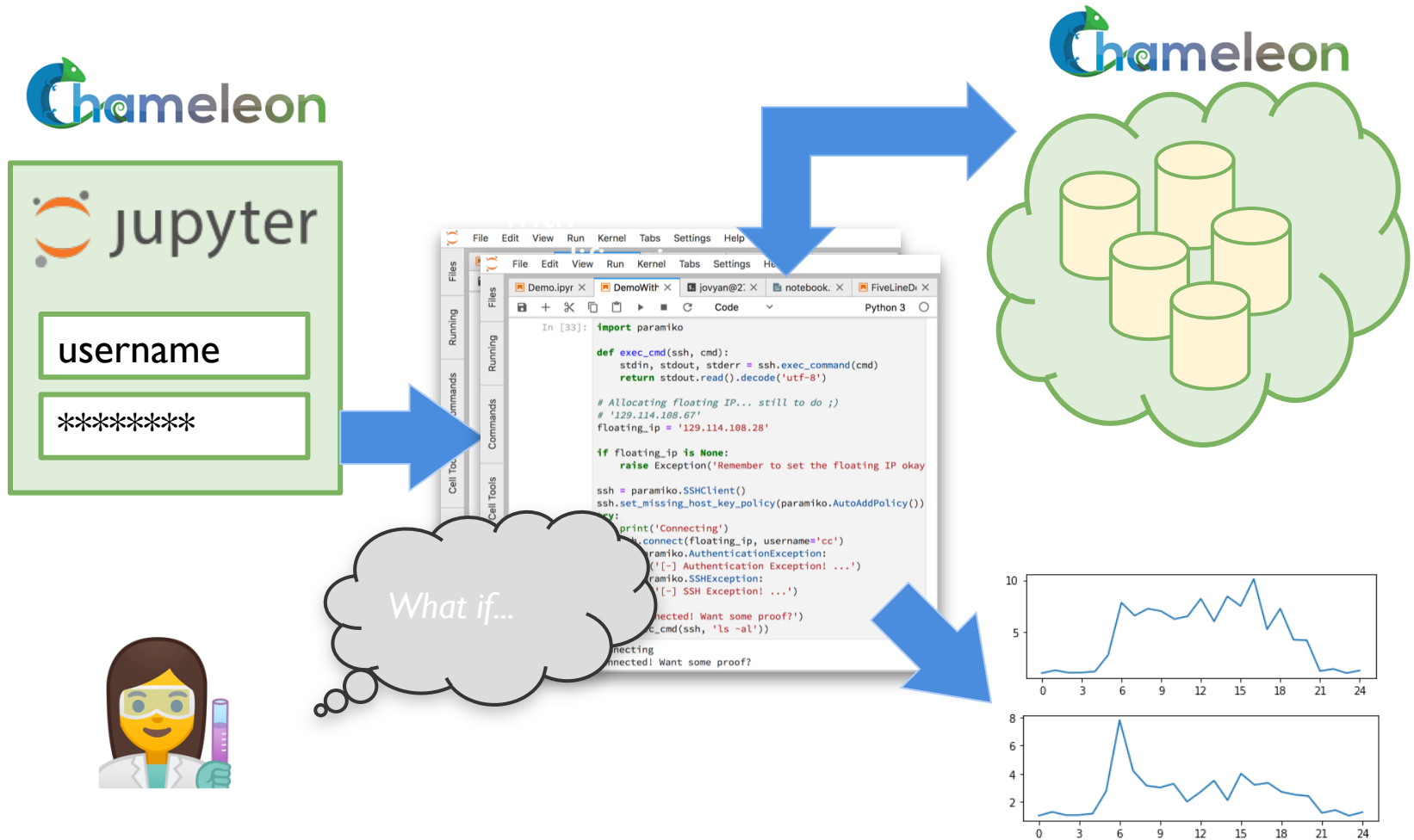
[19]: from chi.ssh import Remote
import time

cross_tacc = Remote(ip="129.114.108.104")
cross_tacc.run("sudo pkill -f iperf3")
time.sleep(2)
```

# JUPYTER ON CHAMELEON



# JUPYTER ON CHAMELEON



# PARTING THOUGHTS

- ▶ Physical environment: Chameleon is a rapidly evolving experimental platform
  - ▶ Originally: “Adapts to the needs of your experiment”
  - ▶ But also: “Adapts to the changing research frontier”
- ▶ Ecosystem: a meeting place of users sharing resources and research
  - ▶ Testbeds are more than just experimental platforms
  - ▶ Common/shared platform is a “common denominator” that can eliminate much complexity that goes into systematic experimentation, sharing, and reproducibility
- ▶ Get engaged – come to our User Meeting!
  - ▶ February 6-7, 2019 in Austin, TX
  - ▶ <https://www.chameleoncloud.org/user-meeting-2019/>
  - ▶ Submission deadline is November 30th



[www.chameleoncloud.org](http://www.chameleoncloud.org)

*Questions?*

[www.chameleoncloud.org](http://www.chameleoncloud.org)

keahey@anl.gov