



www.chameleoncloud.org

THE MANY COLORS OF CHAMELEON

Kate Keahey

Mathematics and CS Division, Argonne National Laboratory

CASE, University of Chicago

keahey@anl.gov

February 6, 2019

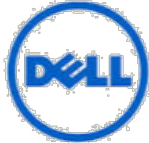
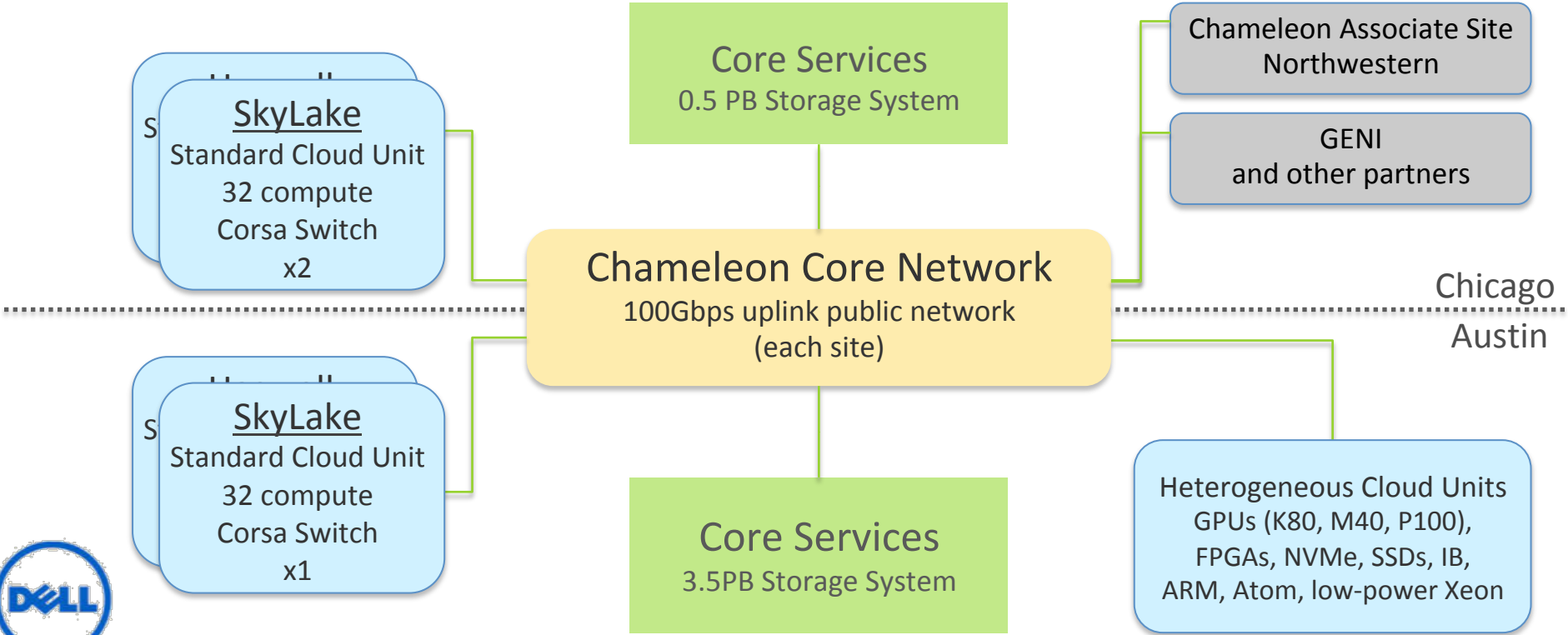
Chameleon User Meeting



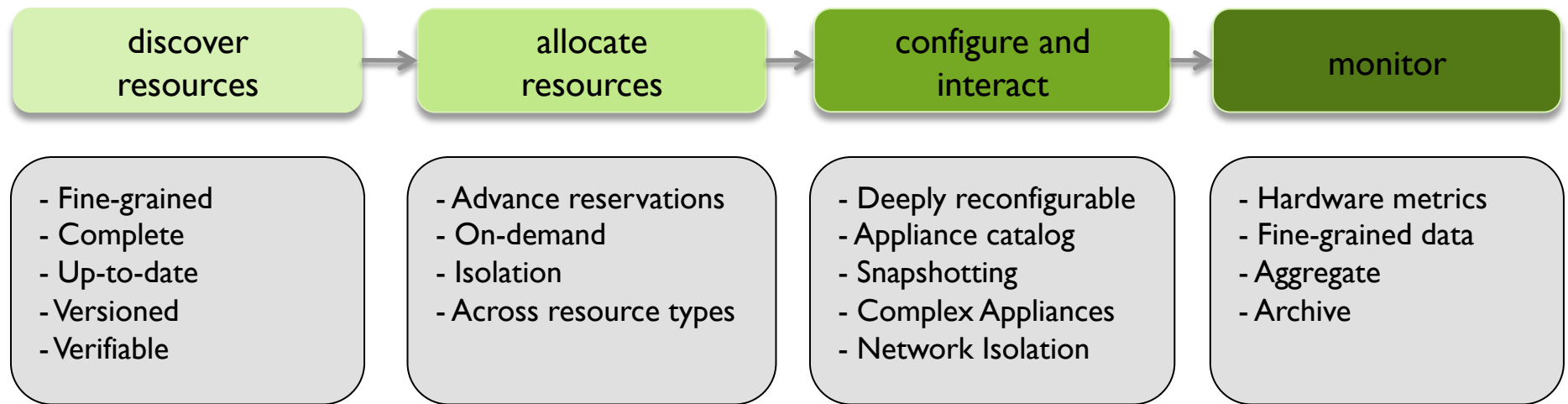
CHAMELEON IN A NUTSHELL

- ▶ We like to change: testbed that adapts itself to your experimental needs
 - ▶ Deep reconfigurability (bare metal) and isolation (CHI) – but also ease of use (KVM)
 - ▶ CHI: power on/off, reboot, custom kernel, serial console access, etc.
- ▶ We want to be all things to all people: balancing large-scale and diverse
 - ▶ Large-scale: ~large homogenous partition (~15,000 cores), 5 PB of storage distributed over 2 sites (now +1!) connected with 100G network...
 - ▶ ...and diverse: ARMs, Atoms, FPGAs, GPUs, Corsix switches, etc.
- ▶ We want to last: cost-effective to deploy, operate, and enhance
 - ▶ Powered by OpenStack with bare metal reconfiguration (Ironic)
 - ▶ Chameleon team contribution recognized as official OpenStack component
- ▶ We live to serve: open, production testbed for Computer Science Research
 - ▶ Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
 - ▶ Currently ~3,000 users, ~500 projects, ~100 institutions

CHAMELEON HARDWARE



EXPERIMENTAL WORKFLOW



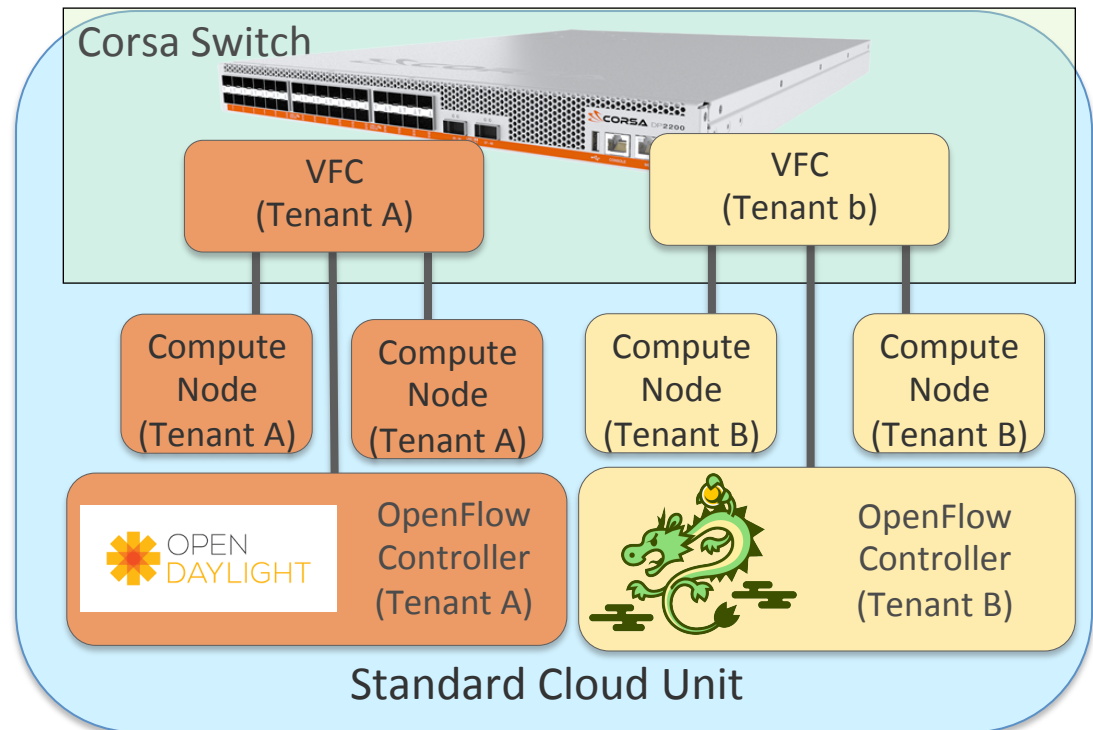
CHI = 65%*OpenStack + 10%*G5K + 25%*”special sauce”

IMPROVING THE PLATFORM: 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)
- ▶ Managing multiple stitches (since Fall 2018)
- ▶ VLAN reservations (since Winter 2019), floating IP reservations coming soon!

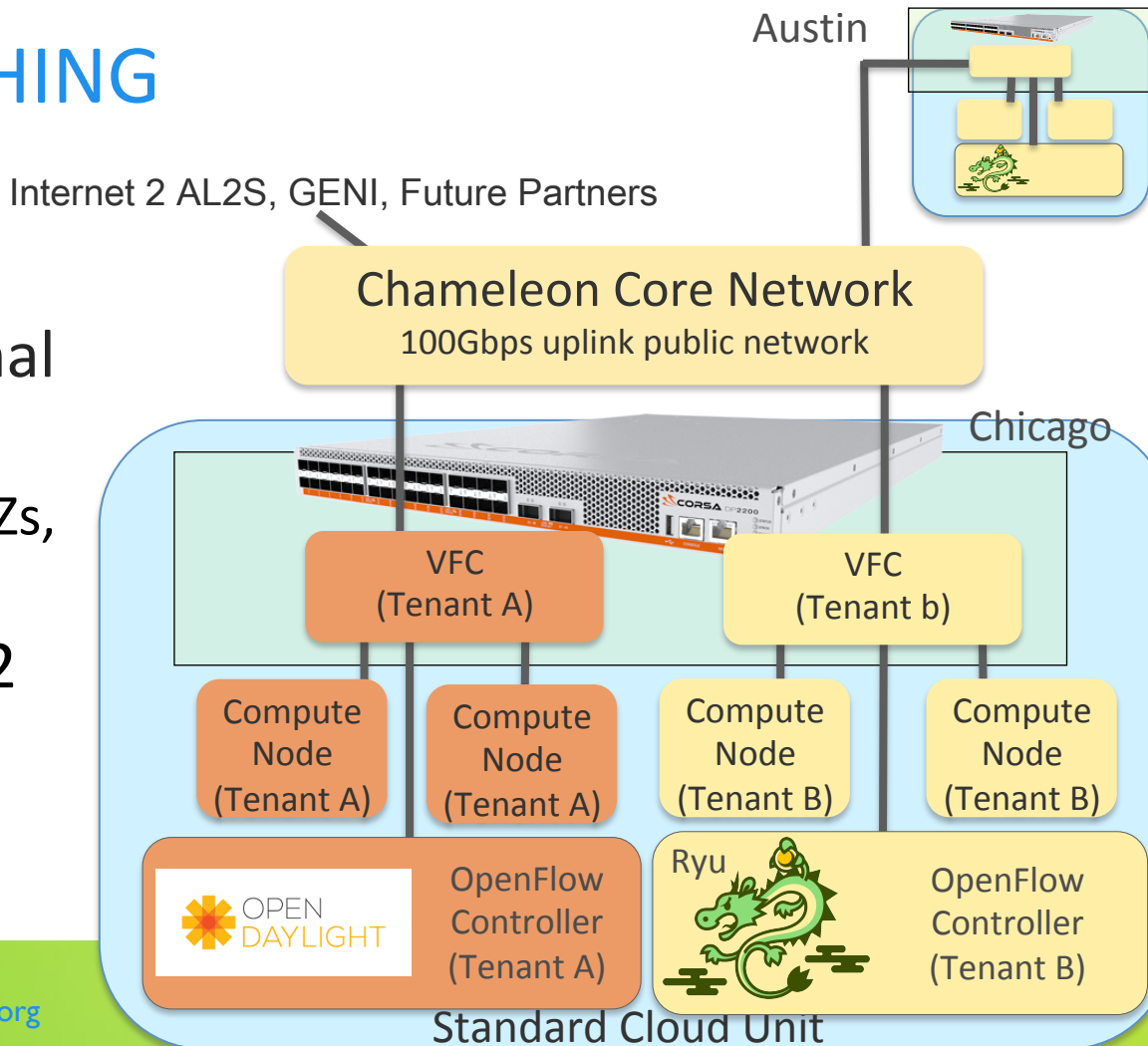
BRING-YOUR-OWN-CONTROLLER (BYOC)

- ▶ Software Defined Networking (SDN)
 - ▶ Corsa Virtual Forwarding Context (VFC)
 - ▶ OpenFlow 1.3
 - ▶ User defined controller
 - ▶ Within Chameleon or anywhere on the Internet
 - ▶ Available on Skylake nodes
- ▶ Supported capabilities
 - ▶ SDN experiments
 - ▶ Experiments requiring non-standard networking capabilities



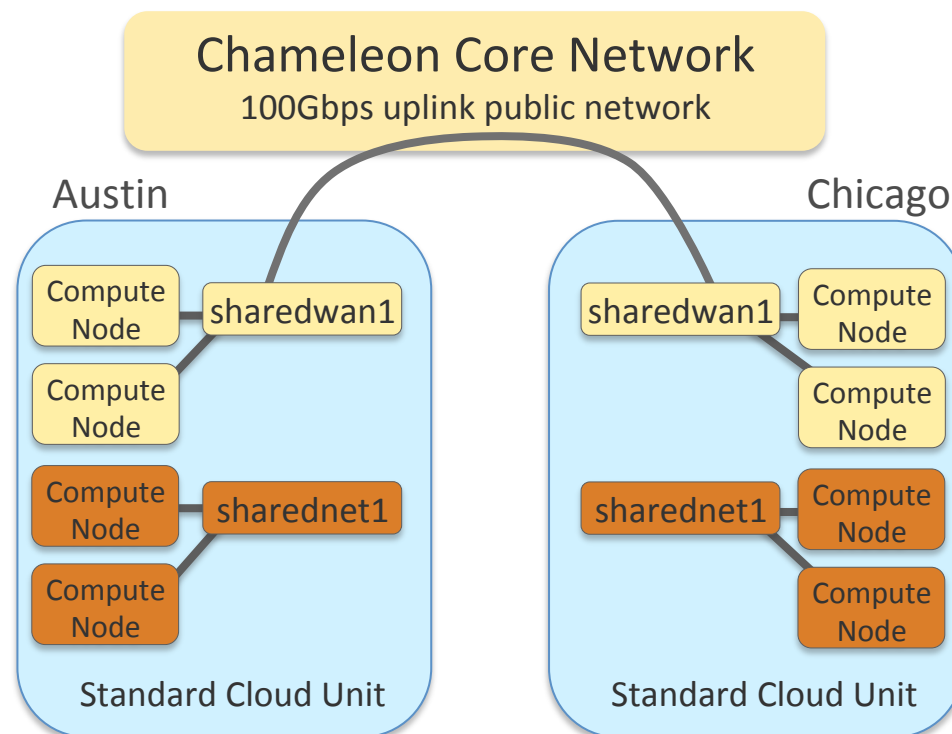
EXTERNAL STITCHING

- ▶ Layer2 VLANs from Chameleon to external partners
 - ▶ ExoGENI, ScienceDMZs, Esnet, and AL2S
- ▶ VFCs with multiple L2 stitched links
 - ▶ Named VFCs



NETWORKING PATTERNS MADE EASY

- ▶ Sharednet1
 - ▶ Pre-configured local shared network
- ▶ Sharedwan1
 - ▶ Stitched shared network
 - ▶ Pre-configured
 - ▶ Connects UC and TACC
 - ▶ Up to 100 Gbps
 - ▶ Ask how to add it to your project!



IMPROVING THE PLATFORM: OTHER FEATURES

- ▶ Lease management: adding/removing nodes to/from a lease, notifications of lease start and impending termination
- ▶ Advance reservation orchestration
- ▶ Power and temperature metrics
- ▶ Whole disk image boot for ARM nodes
- ▶ New appliances (Hadoop, ExoGENI, BYOC examples) and a richer set of appliance features: FUSE module and networking support
- ▶ Usability features: multi-region configuration, single login to all web interfaces, better access to information, better error handling, software self-updates, better appliance publishing, documentation overhaul, etc.
- ▶ Chameleon traces are now available at www.scienceclouds.org

BEYOND THE PLATFORM: BUILDING AN ECOSYSTEM

- ▶ Helping hardware providers interact
 - ▶ Bring Your Own Hardware (BYOH)
 - ▶ CHI-in-a-Box: deploy your own Chameleon site
- ▶ Helping our user interact – with us but primarily with each other
 - ▶ Facilitating contributions of appliances, tools, and other artifacts: appliance catalog, blog as a publishing platform, and eventually notebooks
 - ▶ Integrating tools for experiment management
 - ▶ Making reproducibility easier
- ▶ Improving communication – not just with us but with our users as well

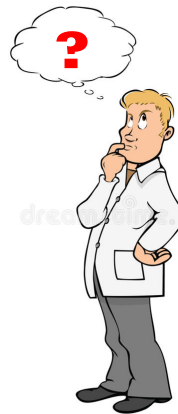
CHI-IN-A-BOX

- ▶ CHI-in-a-box: packaging a commodity-based testbed
 - ▶ First released in summer 2018, continuously improving
- ▶ CHI-in-a-box scenarios
 - ▶ Independent testbed: package assumes independent account/project management, portal, and support
 - ▶ Chameleon extension: join the Chameleon testbed (currently serving only selected users), and includes both user and operations support
 - ▶ Part-time extension: define and implement contribution models
 - ▶ Part-time Chameleon extension: like Chameleon extension but with the option to take the testbed offline for certain time periods (support is limited)
- ▶ Adoption
 - ▶ New Chameleon Associate Site at Northwestern since fall 2018 – new networking!
 - ▶ Two organizations working on independent testbed configuration



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

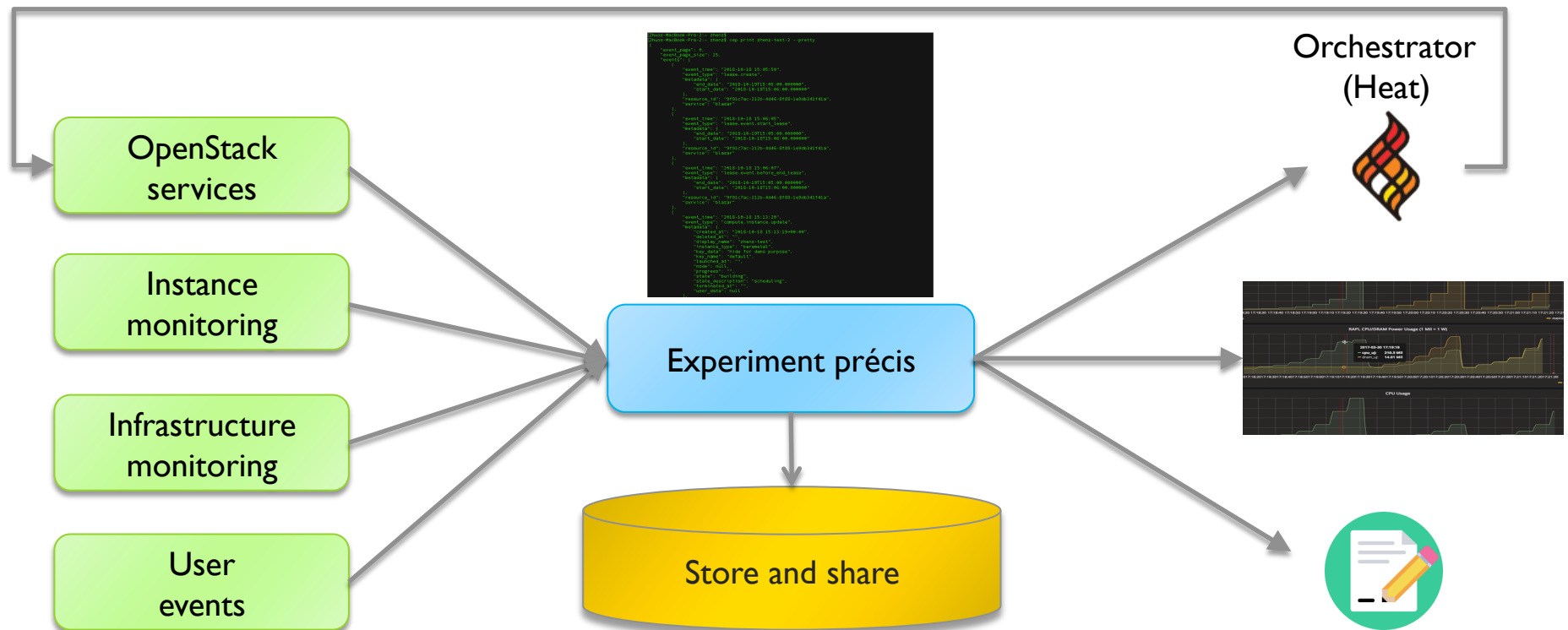
- ▶ Testbed versioning (collaboration with Grid'5000)
 - ▶ Based on representations and tools developed by G5K
 - ▶ >50 versions since public availability – and counting
 - ▶ Still working on: better firmware version management
- ▶ Appliance management
 - ▶ Configuration, versioning, publication
 - ▶ 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

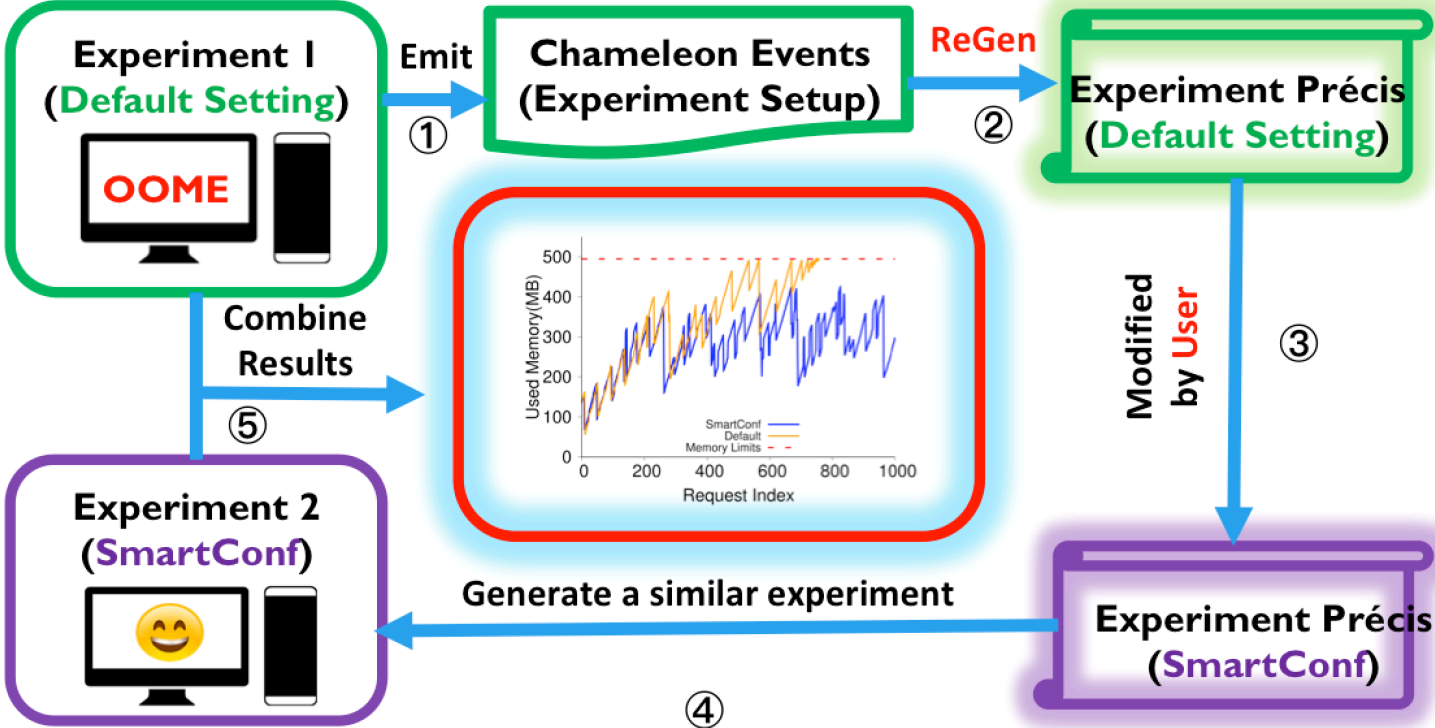
- ▶ Everything in a testbed is a recorded event
- ▶ The resources you used
- ▶ The appliance/image you deployed
- ▶ The monitoring information your experiment generated
- ▶ Plus any information you choose to share with us: e.g., “start power_exp_23” and “stop power_exp_23”

-
- ▶ Experiment précis: information about your experiment made available in a “consumable” form

REPEATABILITY: EXPERIMENT PRÉCIS



EXPERIMENT PRÉCIS: A CASE STUDY

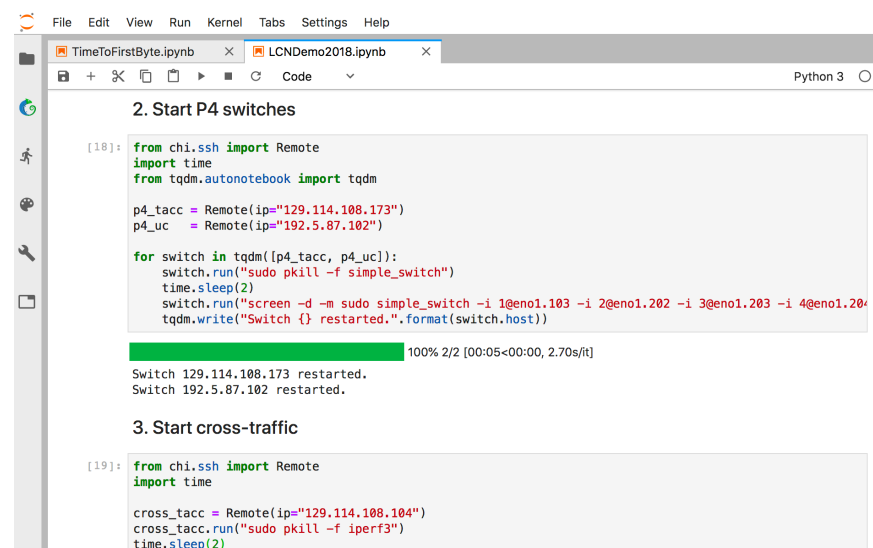


INTERACTIVE PAPERS

- ▶ What does it mean to document a process?
- ▶ Some 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

CHAMELEON JUPYTER INTEGRATION

- ▶ Combining the ease of notebooks and the power of a shared platform
 - ▶ Storytelling with Jupyter: ideas/text, process/code, results
 - ▶ Chameleon shared experimental platform
- ▶ JupyterLab server for our users
 - ▶ Just go to jupyter.chameleoncloud.org and log in with your Chameleon credentials
- ▶ Chameleon/Jupyter integration
 - ▶ Alternative interface
 - ▶ All the main testbed functions
 - ▶ “Hello World” template



```
File Edit View Run Kernel Tabs Settings Help
TimeToFirstByte.ipynb LCNDEmo2018.ipynb Python 3
2. Start P4 switches
[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)
```

Screencast of a complex experiment: <https://vimeo.com/297210055>

SHARING, EXPERIMENTING, LEVERAGING

- ▶ Sharing Jupyter notebooks in Chameleon
 - ▶ Today: from home directory to sharing via our Swift storage with your project members
 - ▶ Challenges ahead: more flexible sharing policy implementation, integrating with github for better versioning and sharing support
- ▶ Automating experiments with Jupyter

PARTING THOUGHTS

- ▶ Physical environment: Chameleon is a rapidly evolving experimental platform
 - ▶ Originally: “Adapts to the needs of your experiment”
 - ▶ Now also: “Adapts to the needs of its community and the changing research frontier”
- ▶ Towards an Ecosystem: a meeting place of users and providers 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
- ▶ Be part of the change: tell us what capabilities we should provide to help you share and leverage the contributions of others!