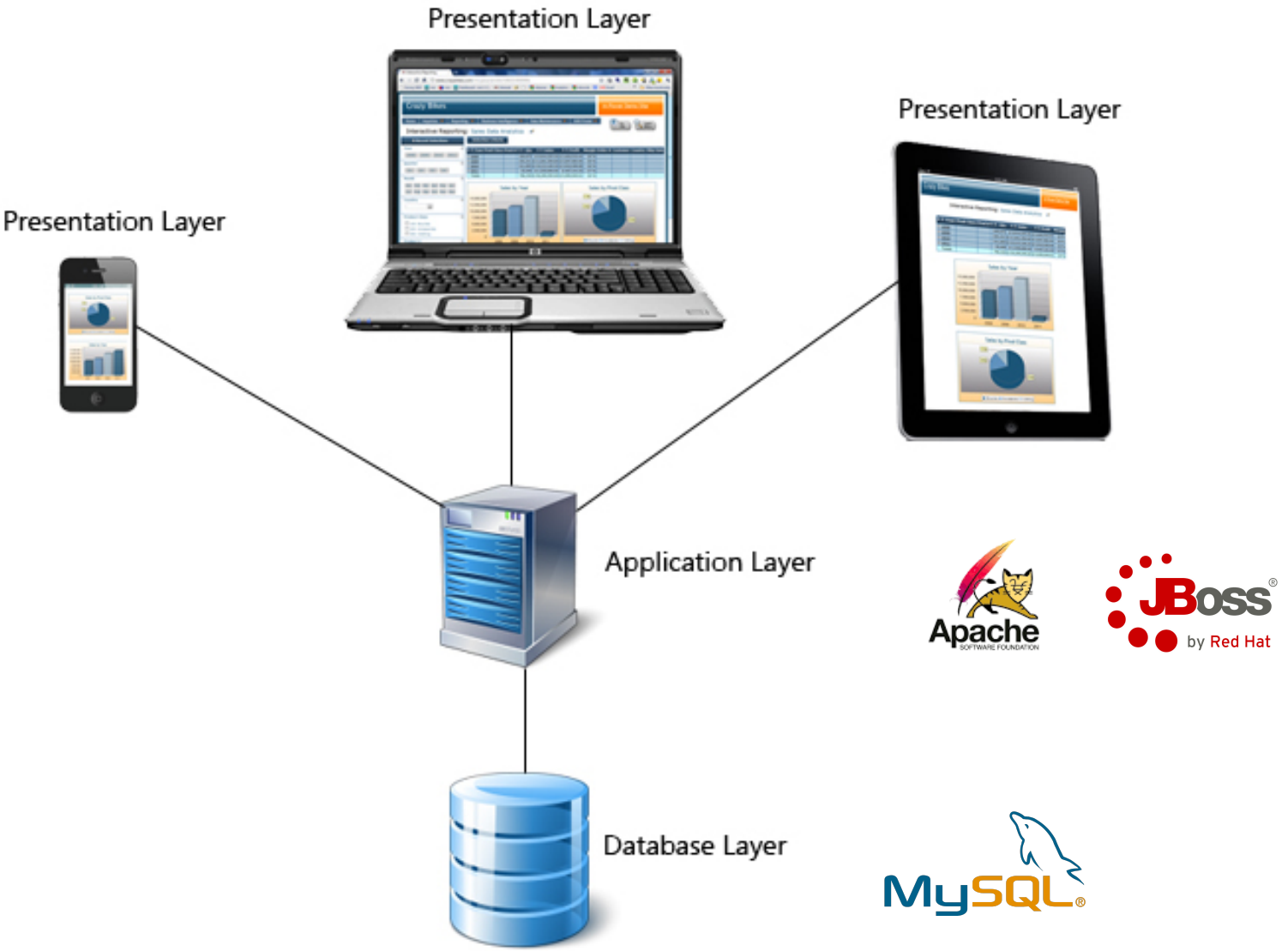
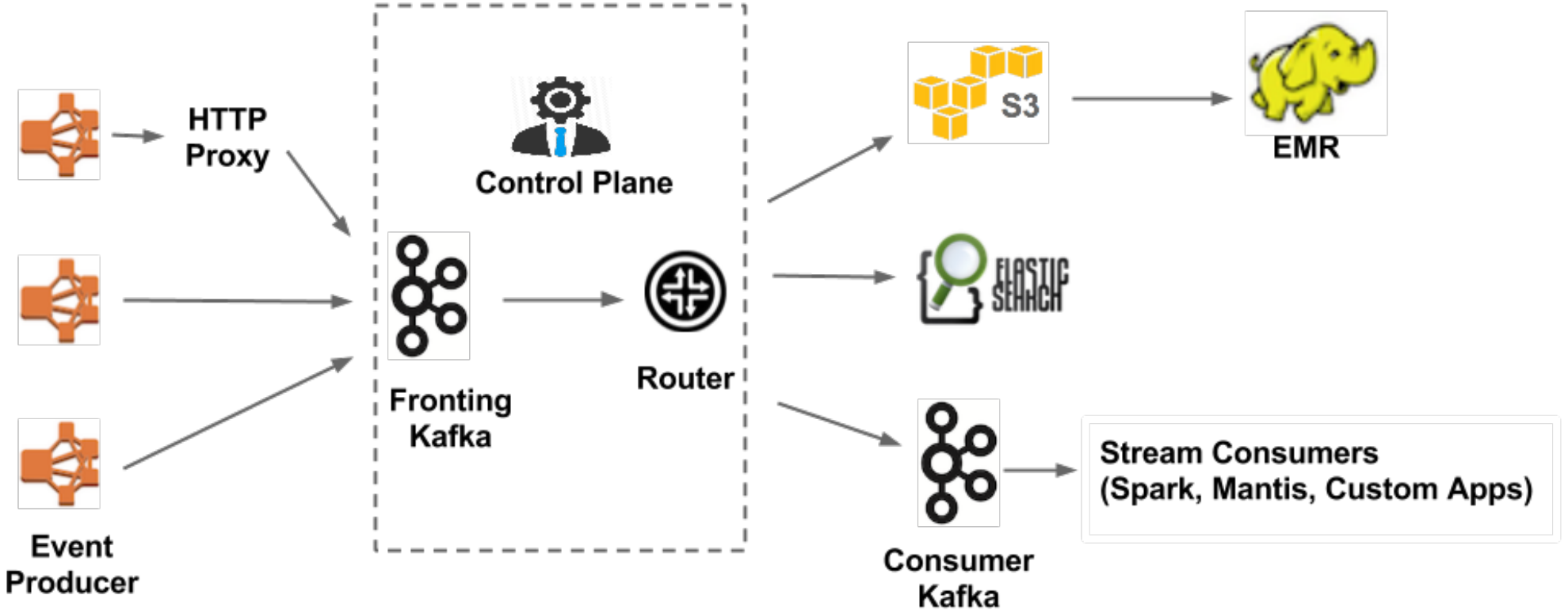
System Deployment & Benchmarking

A “Simple” Application

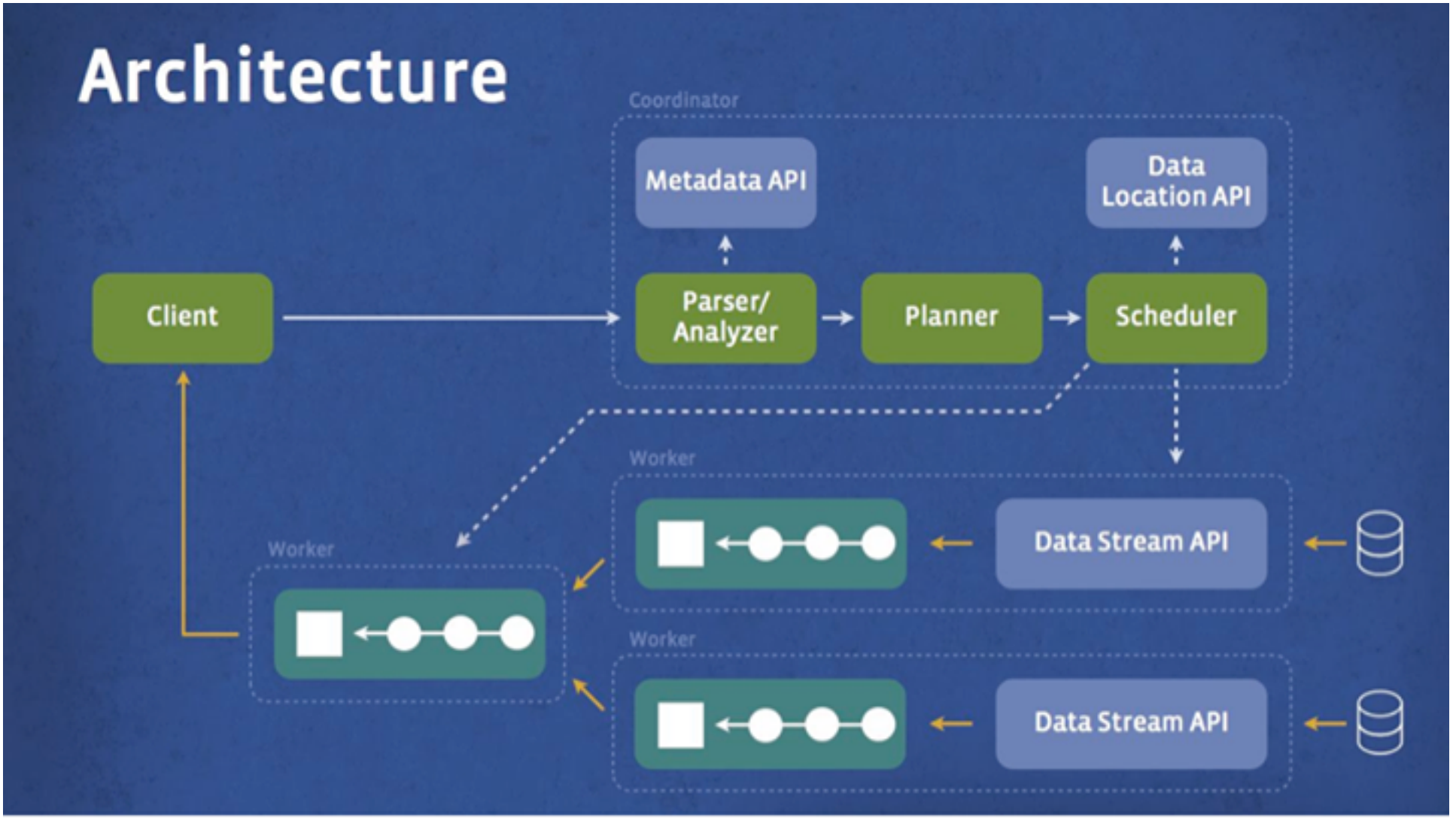


Complex Applications (Netflix)



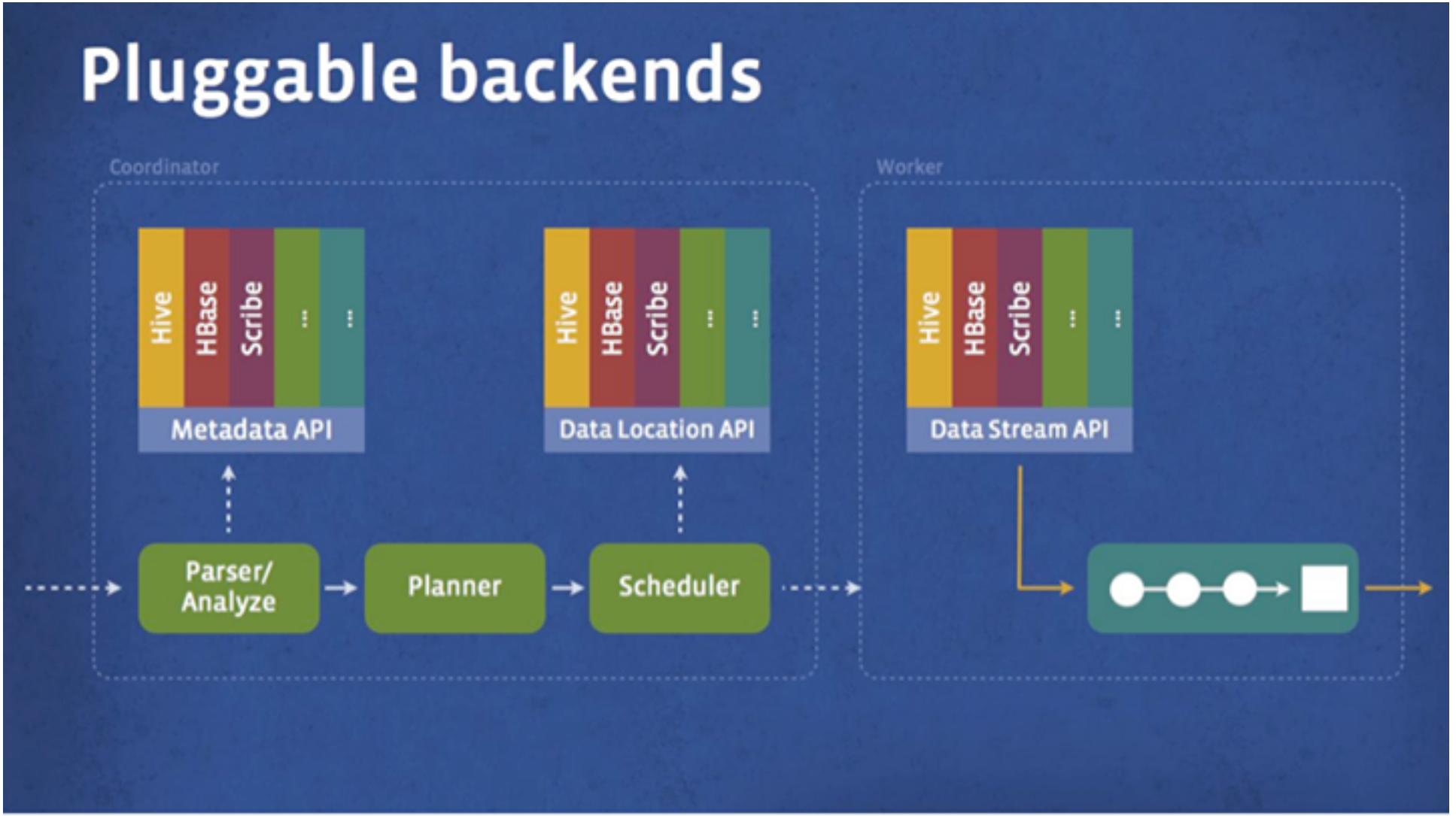
Source: https://medium.com/netflix-techblog/evolution-of-the-netflix-data-pipeline-da246ca36905

Complex Applications (Facebook)



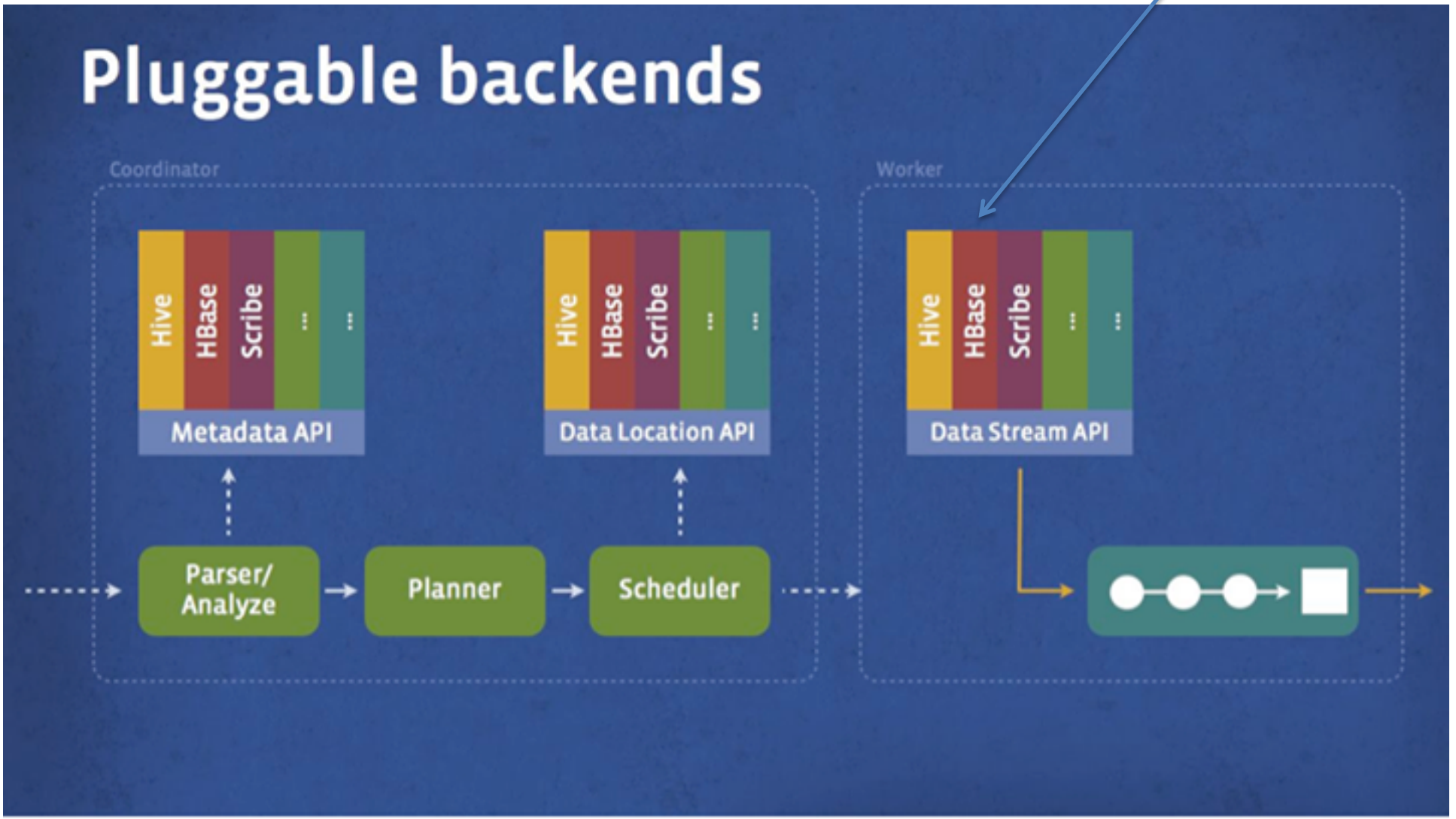
Source: https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920/?s=keen-io

Complex Applications (Facebook)



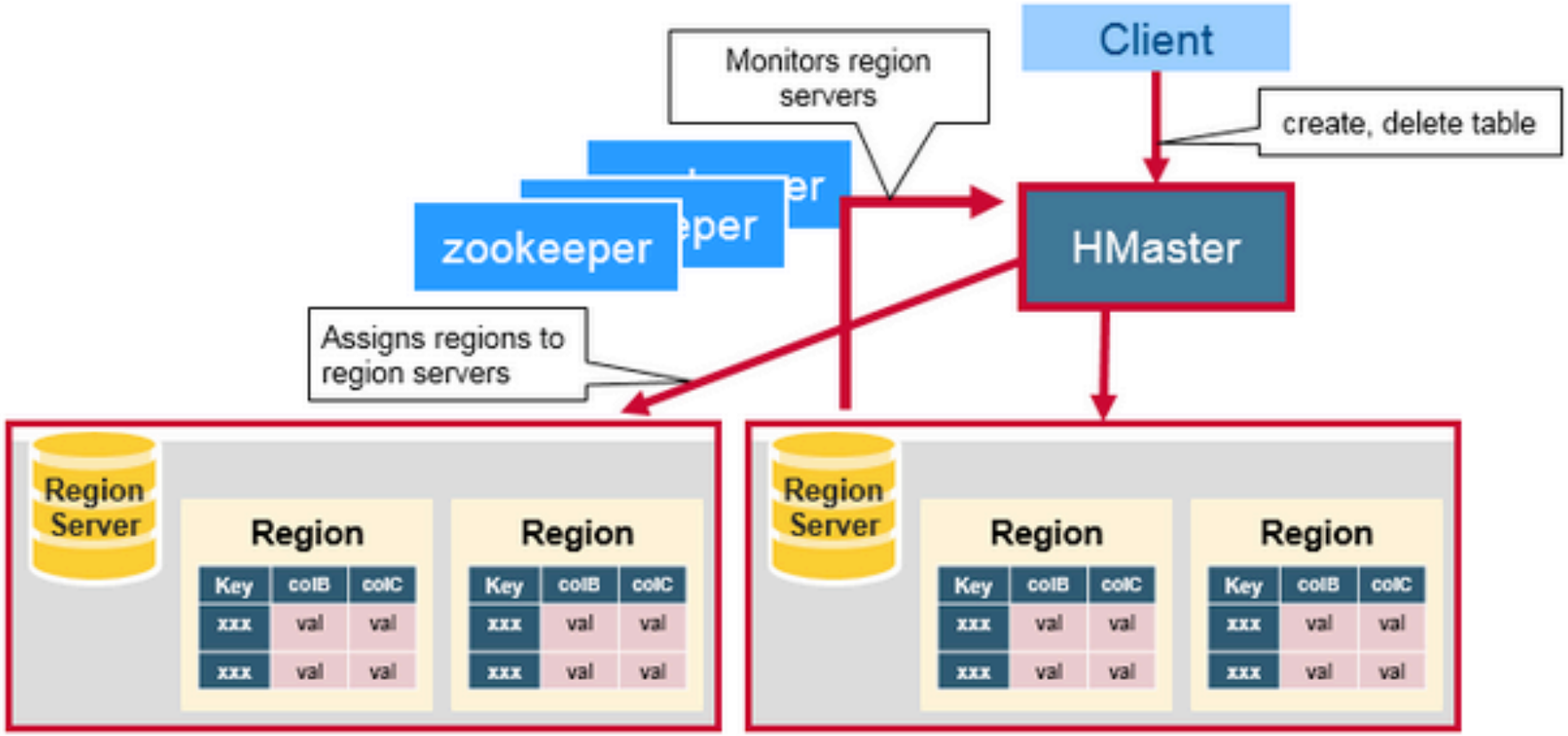
Source: https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920/?s=keen-io

Complex Applications (Facebook)

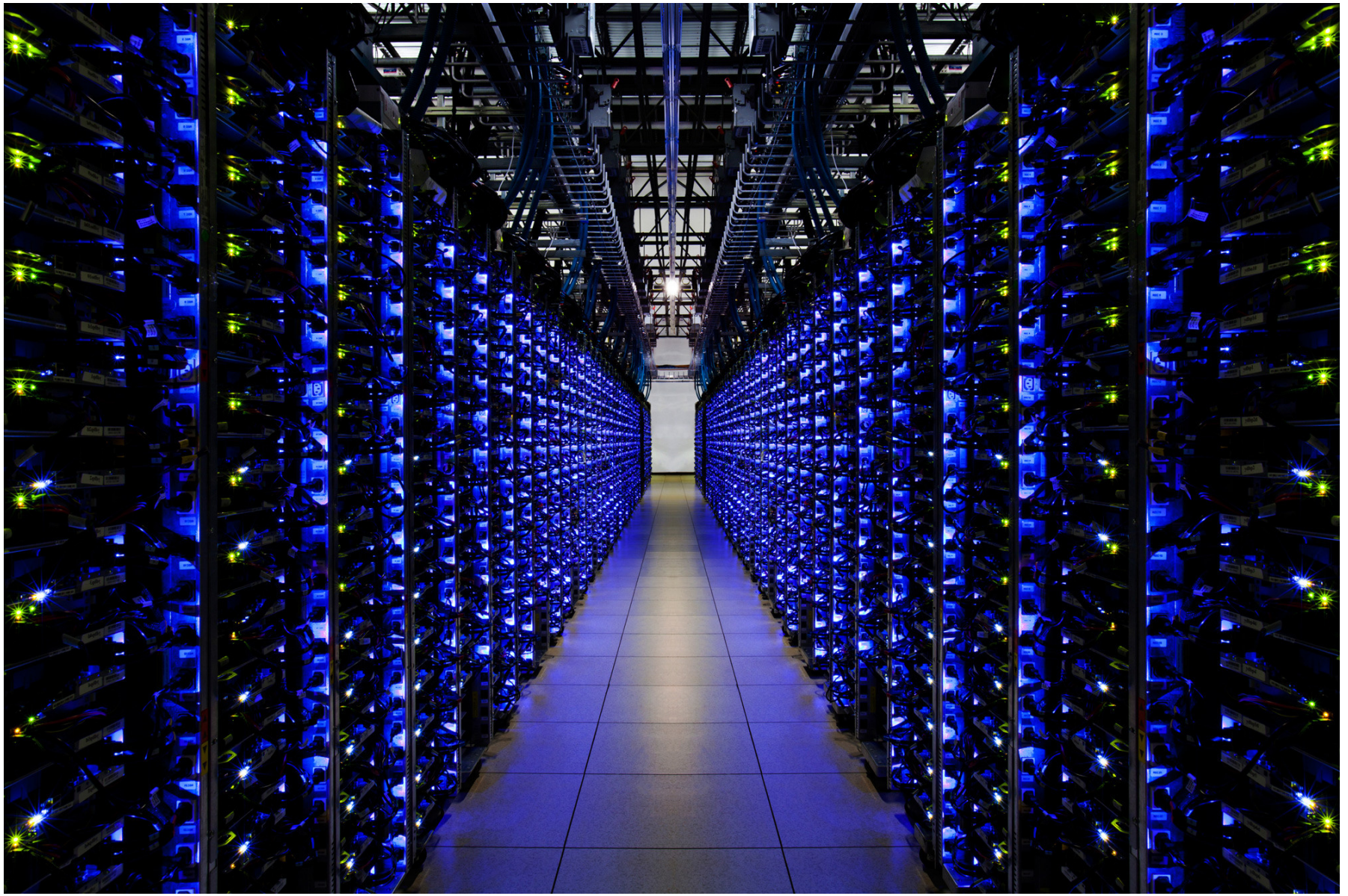


Source: https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920/?s=keen-io

Complex Applications (HBase)



Complex Infrastructures



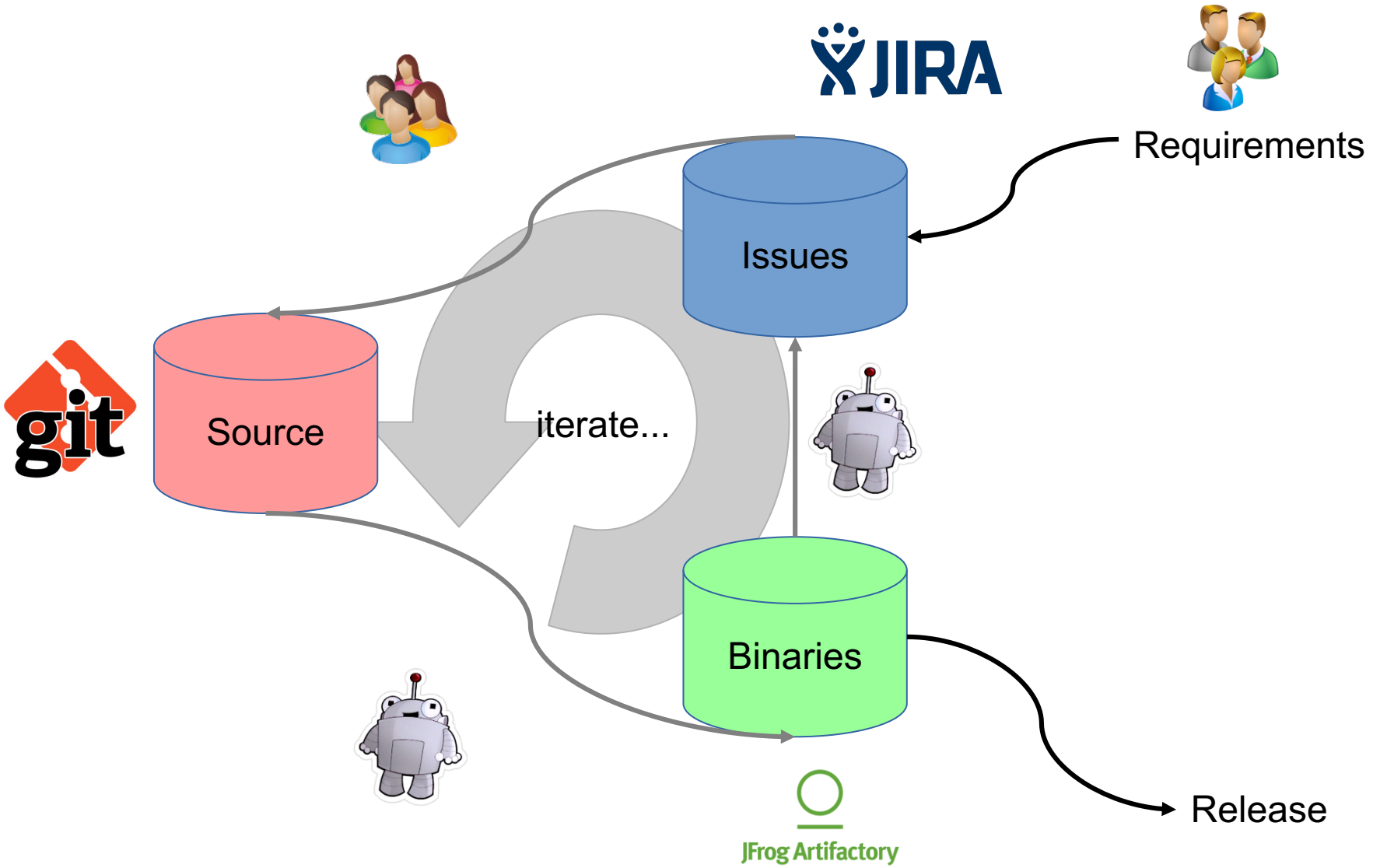
Challenges

- Deployment
  - Multiple components with specific requirements
  - Heterogeneous hardware, software and services (e.g., cloud computing services)

- Configuration
  - Specific to each software/hardware component
  - Optimal configurations will change with time...
- Monitoring and Benchmarking
  - Finding anomalies (performance, failures, ...) in complex applications and infrastructures

Yet another example...

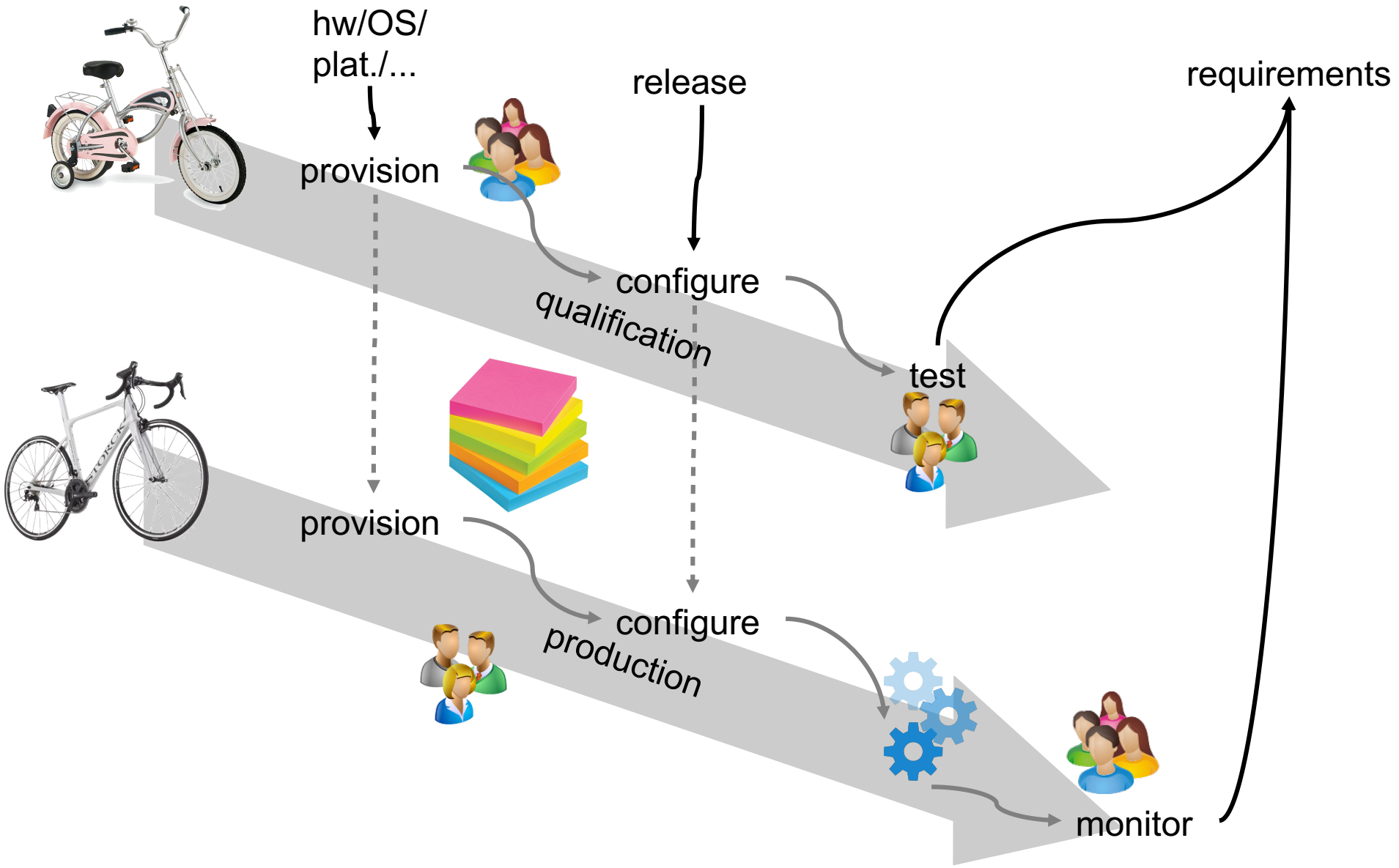
Agile development



Agile developement

- Rests on formal (versioned) source, binary and documentation repositories
  - Unambigous current state
  - Ability to back track
- Automated build and test
  - Fast feedback
- Fast iterations
- Quick reaction to frequent small changes

Operations


hw/OS/
plat./...
release
requirements
provision
configure
qualification
test
provision
configure
production
monitor

Operations

- Manual provisioning and configuration
- Informal communication between qualification and production stages
- Consequences:
  - Not repeatable or reproduceable
  - Configuration drift
  - “Snowflake” servers (don't even look at it…)
  - Subjective monitoring

Infrastructure as Code

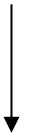
- Hardware provisioning with scripts
  - In contrast to: physically unboxing and plugging
- Software provisioning with scripts
  - In contrast to: clicking through setup wizards
- Configuration with scripts
  - In contrast to: clicking through control panels

- Include infrastructure scripts in agile process!

Agile development

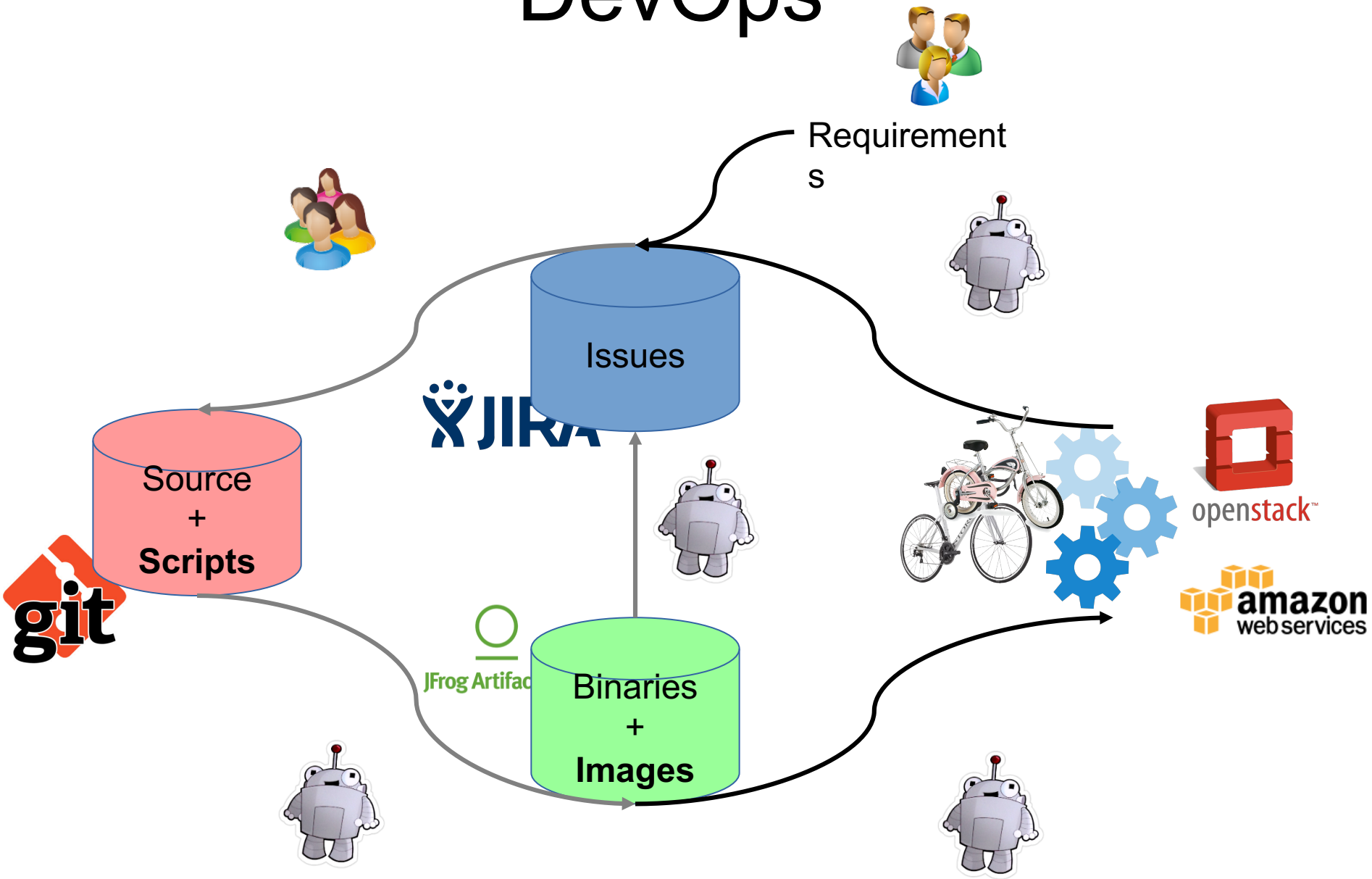
+

Infrastructure as Code



DevOps

DevOps


Requirements
Issues
JIRA
Source
+
Scripts
git
JFrog Artifac
Binaries
+
Images
openstack
amazon
web services

DevOps

- Development and deployment are self-documenting and versioned
- Deployment is reproduceable and repeatable
- Servers are disposable and consistent

- Supports fast, frequent, small changes!

Roadmap

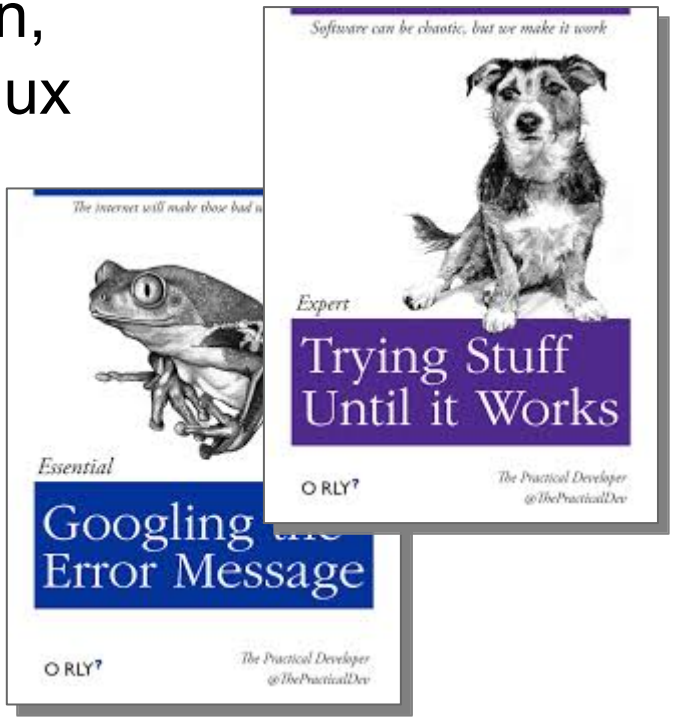
- Distributed systems architectures
  - Common distribution patterns
- Infrastructure automation
  - Virtualization and containers
  - Cloud Services
- Platform automation
  - Provisioning
- Monitoring and evaluation
  - Log collection and analysis
  - Benchmarking

Assessment

- Project (50%) – minimum grade: 8 values
  - Phase 1: 20/11/2020
  - Phase 2: 15/01/2021
  - Presentation: 27/01/2021
- Written exam (50%) – minimum grade: 8 values
  - 06/01/2021

Main references

- Kief Morris, *Infrastructure as Code - Managing Servers in the Cloud*, O'Reilly, 2016.
- *Raj Jain, The Art of Computer Systems Performance Analysis*,
  Wiley, 1991.
- Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin. UNIX and Linux System Administration Handbook (5th Edition), Addison-Wesley Professional, 2017.
- Sam R. Alapati, Modern Linux Administration, O'Reilly (early release)



Further reading

- Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. 2003. Xen and the art of virtualization. SIGOPS Oper. Syst. Rev. 37, 5 (October 2003), 164-177. DOI: https://doi.org/10.1145/1165389.945462
- Neves F, Machado N, Pereira JO. 2018. Falcon: A Practical Log-based Analysis Tool for Distributed Systems. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN18).
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A view of cloud computing. Commun. ACM 53, 4 (April 2010), 50-58. DOI: https://doi.org/10.1145/1721654.1721672
- Chunqiang Tang, Thawan Kooburat, Pradeep Venkatachalam, Akshay Chander, Zhe Wen, Aravind Narayanan, Patrick Dowell, and Robert Karl. 2015. Holistic configuration management at Facebook. In Proceedings of the 25th Symposium on Operating Systems Principles (SOSP '15). ACM, New York, NY, USA, 328-343. DOI: https://doi.org/10.1145/2815400.2815401
- J. Sahoo, S. Mohapatra and R. Lath, "Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues," *2010 Second International Conference on Computer and Network Technology*, Bangkok, 2010, pp. 222-226.

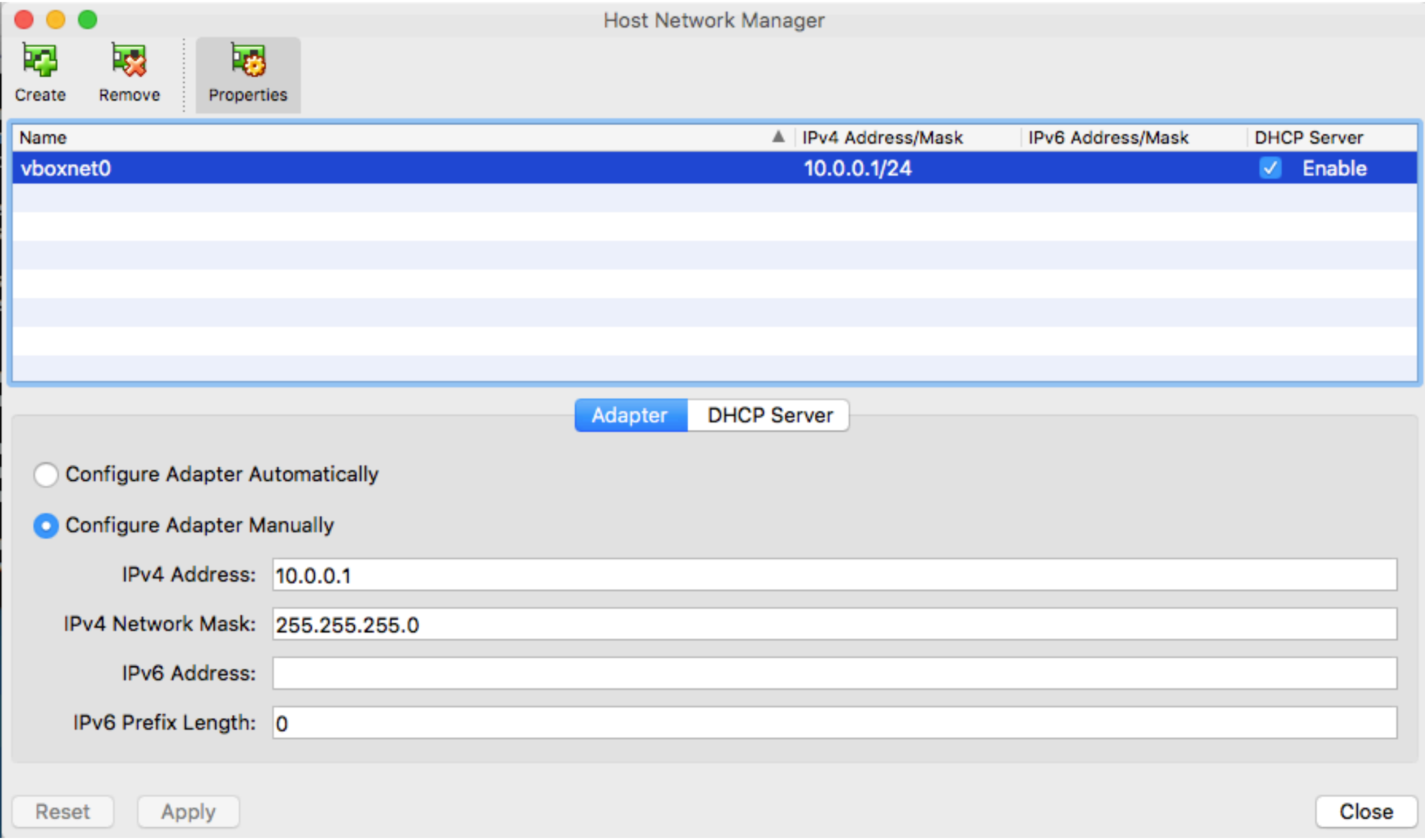
Team

- João Paulo - jtpaulo@di.uminho.pt

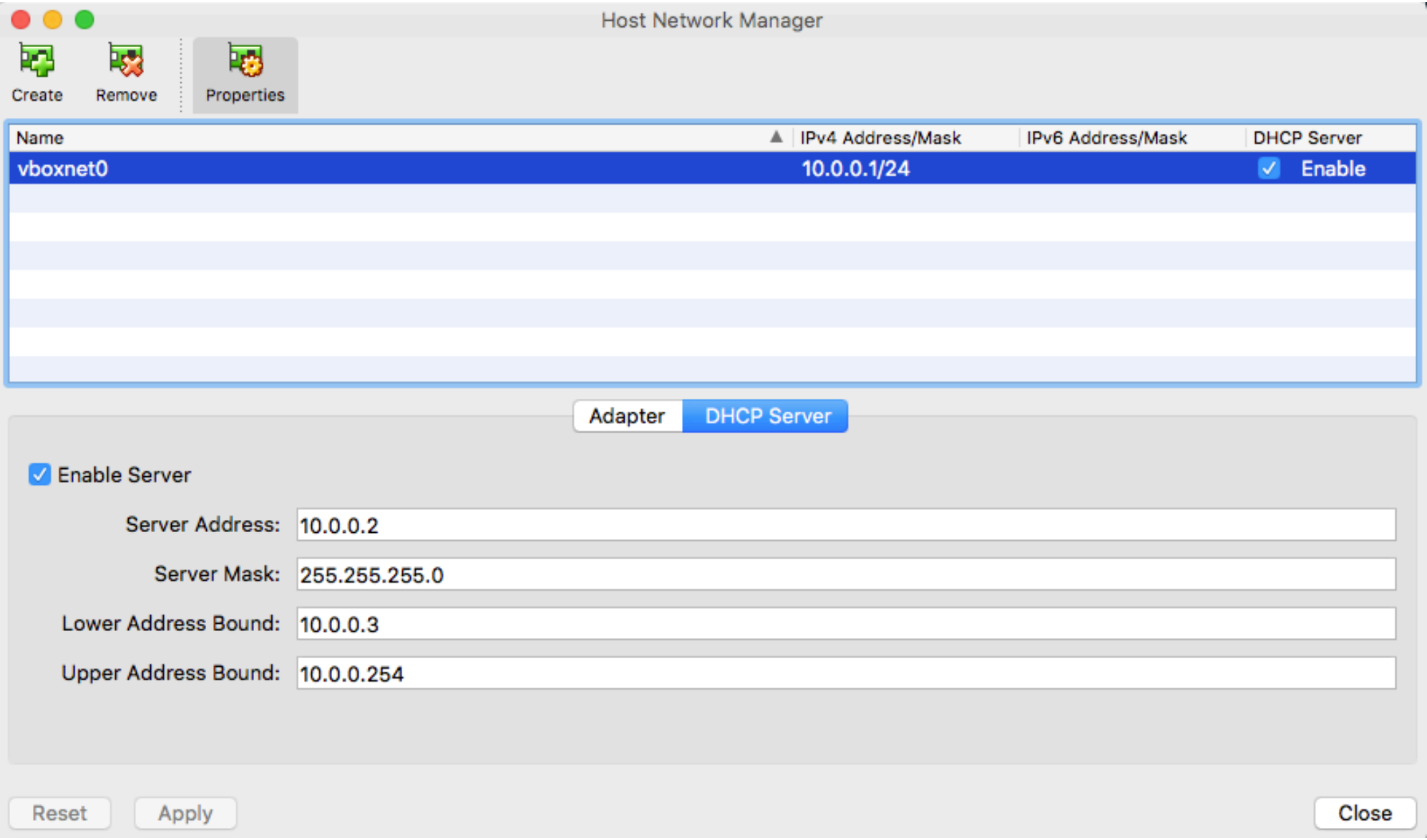
Warm-up

- Install VirtualBox 6.1 (https://www.virtualbox.org)
- Configure Host-only network
  - File -> Host Network Manager (Mac OSX)

Warm-up

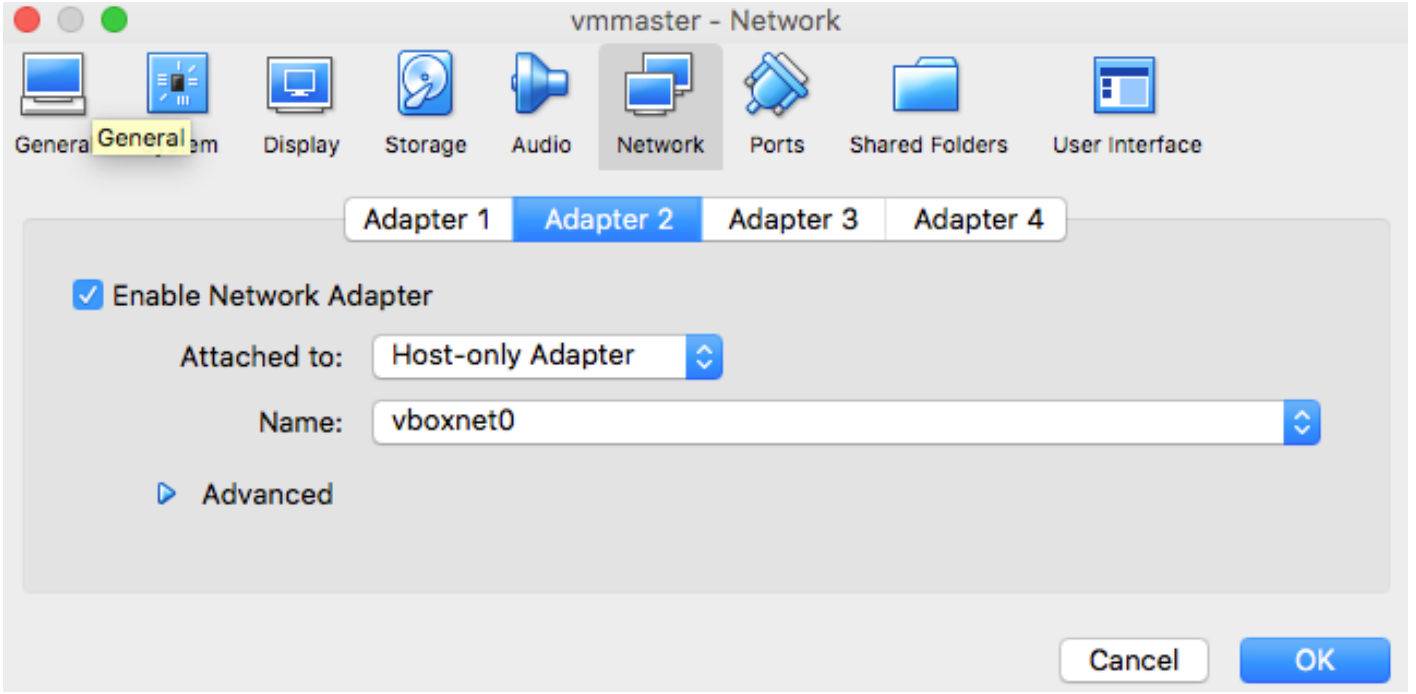

Host Network Manager
Create
Remove
Properties
Name
IPv4 Address/Mask
IPv6 Address/Mask
DHCP Server
vboxnet0
10.0.0.1/24
Enable
Adapter
DHCP Server
Configure Adapter Automatically
Configure Adapter Manually
IPv4 Address:
10.0.0.1
IPv4 Network Mask:
255.255.255.0
IPv6 Address:
IPv6 Prefix Length:
0
Reset
Apply
Close

Warm-up


Host Network Manager
Create
Remove
Properties
Name
IPv4 Address/Mask
IPv6 Address/Mask
DHCP Server
vboxnet0
10.0.0.1/24
Enable
Adapter
DHCP Server
Enable Server
Server Address:
10.0.0.2
Server Mask:
255.255.255.0
Lower Address Bound:
10.0.0.3
Upper Address Bound:
10.0.0.254
Reset
Apply
Close

Warm-up

- Setup one Ubuntu (64-bit) virtual machine
- Plan CPU, RAM (1GB) and Disk (10GB) resources so that multiple VMs can be deployed at the same host
  - VDI disk
  - Configure an additional network adapter (settings)



Warm-up

- Install Ubuntu from ISO file (Ubuntu 20.04 server)
  - Install OpenSSH server package
- Launch the VM
  - “ip add” command lists the available network interfaces
  - Configure network (/etc/netplan/)

```
enp0s8:
    addresses:
        - 10.0.0.3/24
```

  - “sudo netplan apply”

Warm-up

- Test SSH from host to the VM
- Configure SSH key access (/etc/ssh/sshd_config)
  - chmod 700 .ssh
  - chmod 600 .ssh/authorized_keys
  - sudo systemctl restart sshd

Warm-up

- Clone the VM to create a new one
  - Generate new MAC for the Network Adapters
- Launch Second VM and configure network (/etc/netplan-> addresses 10.0.0.4)
- Reboot VM
- Test connection between VMs and Host