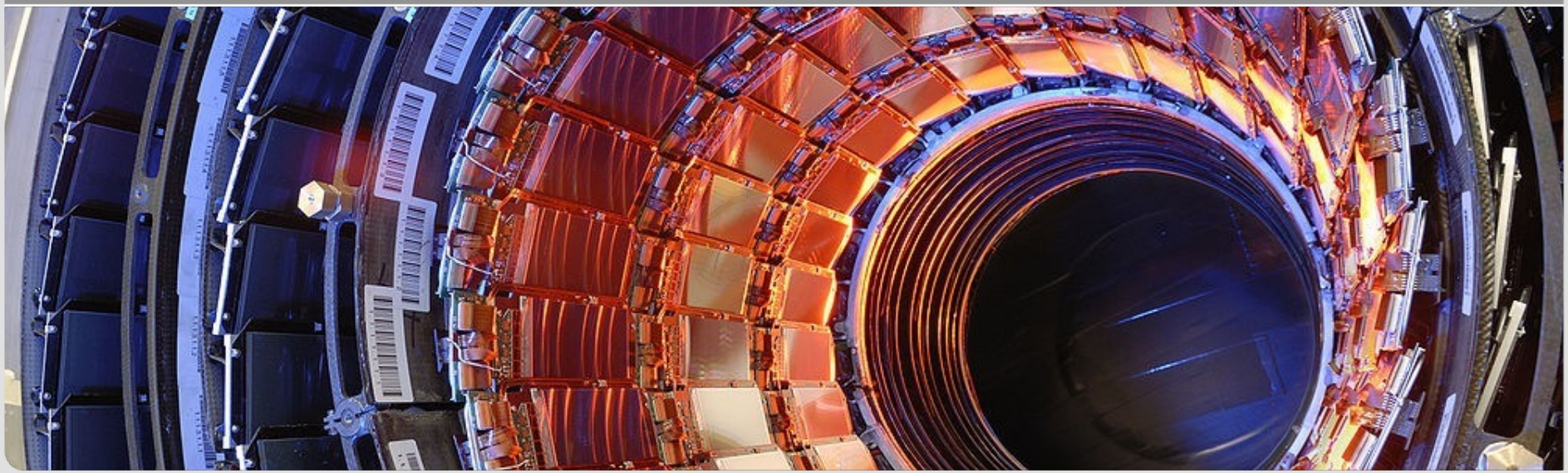
Karlsruhe Institute of Technology

Active Job Monitoring in Pilots

Eileen Kuehn, Max Fischer, **Manuel Giffels**, Christopher Jung, Andreas Petzold

Steinbuch Centre for Computing, GridKa



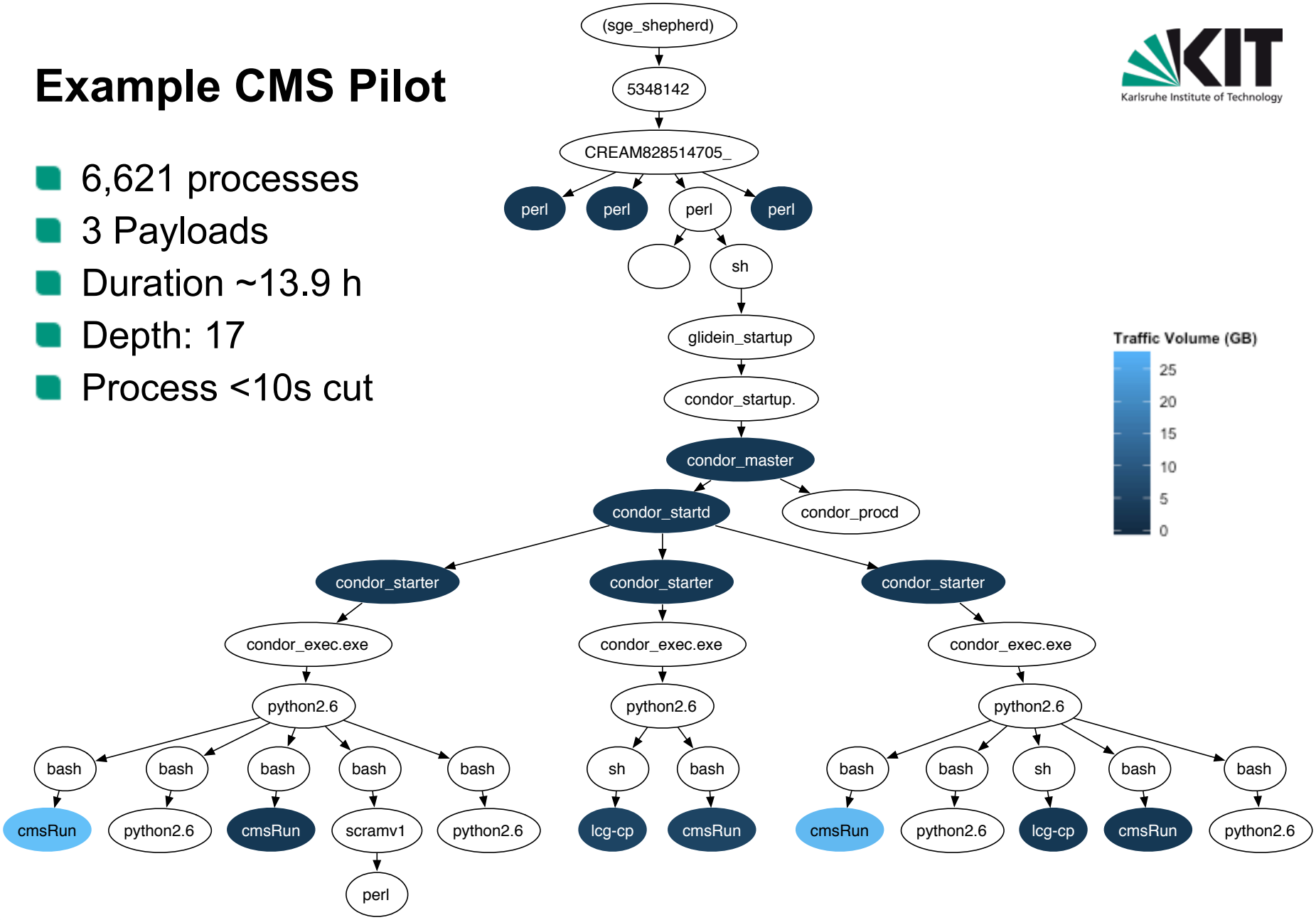
Motivation

- Todays WLCG batch system usage dominated by pilots
  - Remote-controlled scheduling by using virtual overlay batch systems
  - Single pilots execute an arbitrary number of payloads
- No monitoring of payload resources via batch system
  - Pilot shields underlying information
  - No existing tools available
- Pilots in batch systems forbidden in other communities

- Current Issues
  - No information about single payloads
  - Efficiency per payload cannot be determined
  - Detection of misbehaving payloads not possible

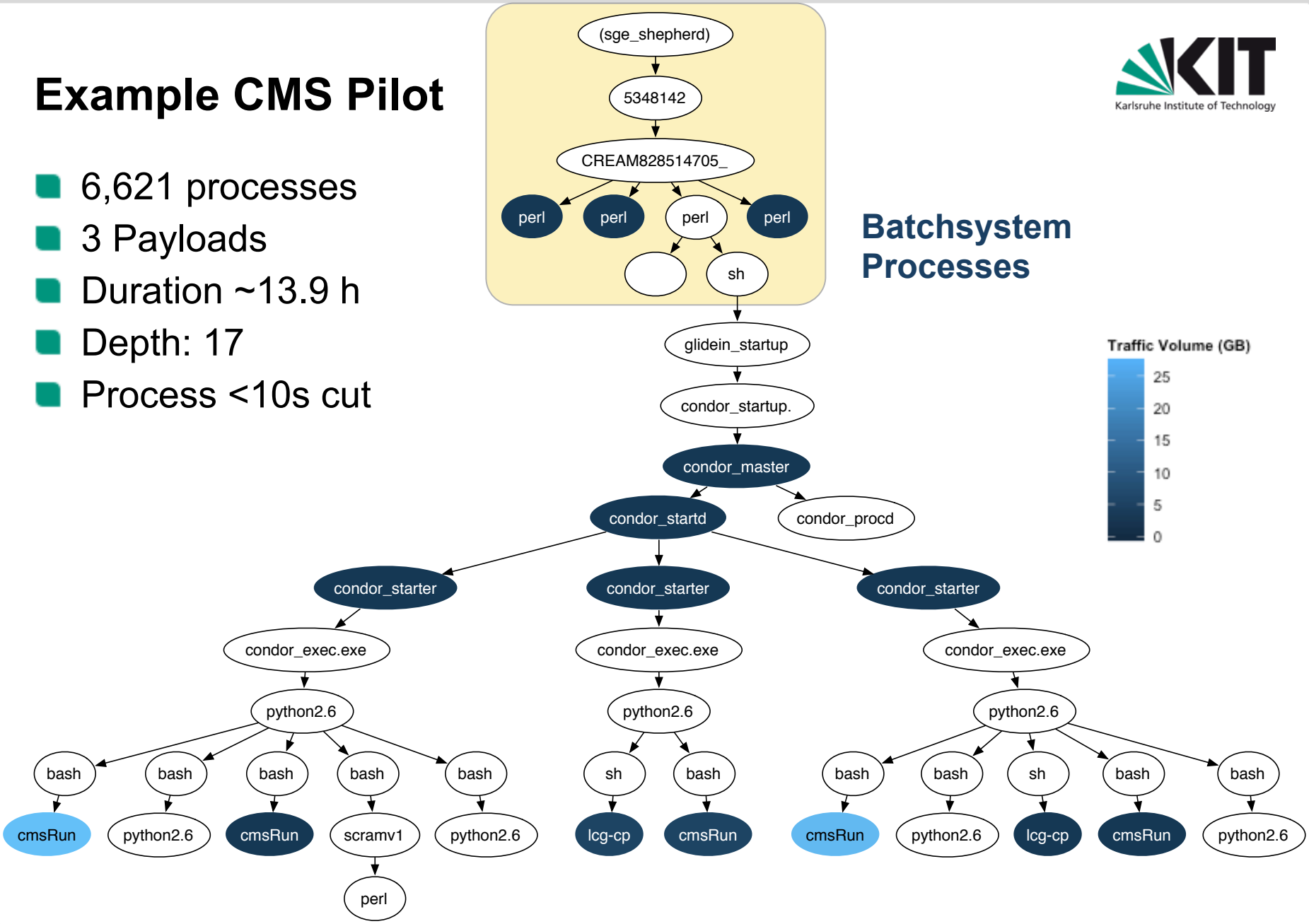
Example CMS Pilot

- 6,621 processes
- 3 Payloads
- Duration ~13.9 h
- Depth: 17
- Process <10s cut



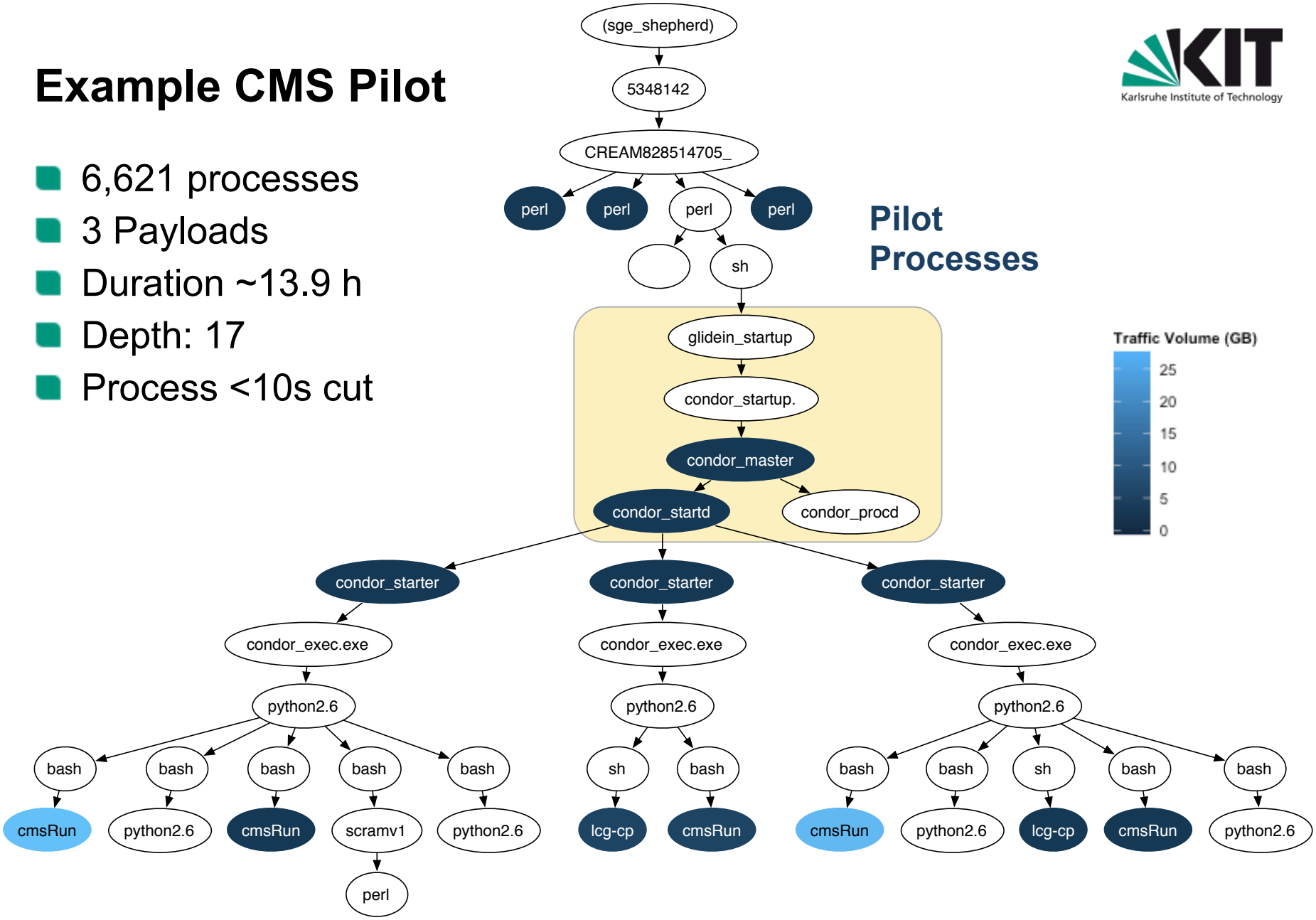
Example CMS Pilot

- 6,621 processes
- 3 Payloads
- Duration ~13.9 h
- Depth: 17
- Process <10s cut



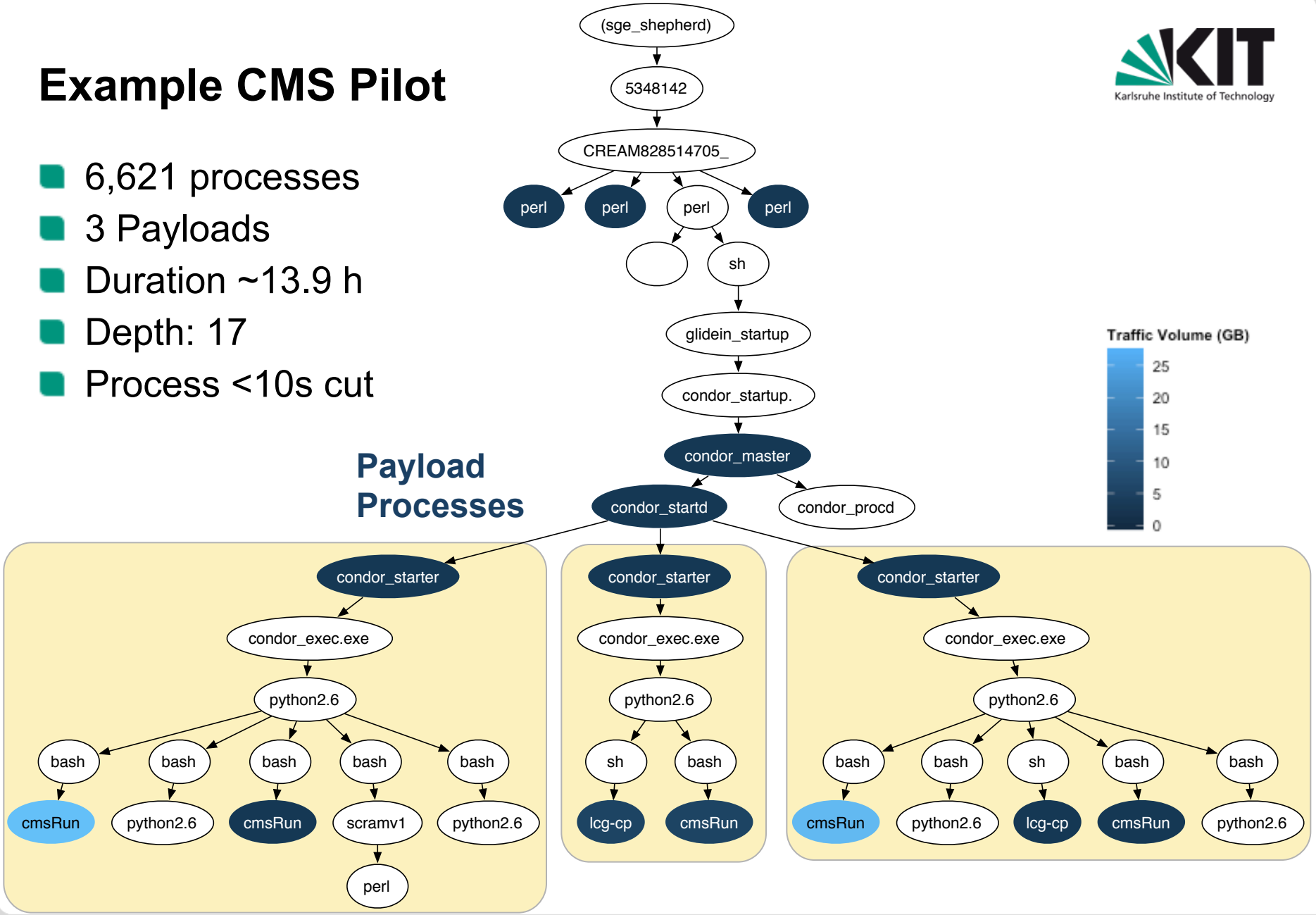
Example CMS Pilot

- 6,621 processes
- 3 Payloads
- Duration ~13.9 h
- Depth: 17
- Process <10s cut



Example CMS Pilot

- 6,621 processes
- 3 Payloads
- Duration ~13.9 h
- Depth: 17
- Process <10s cut



Payload Monitoring Tool

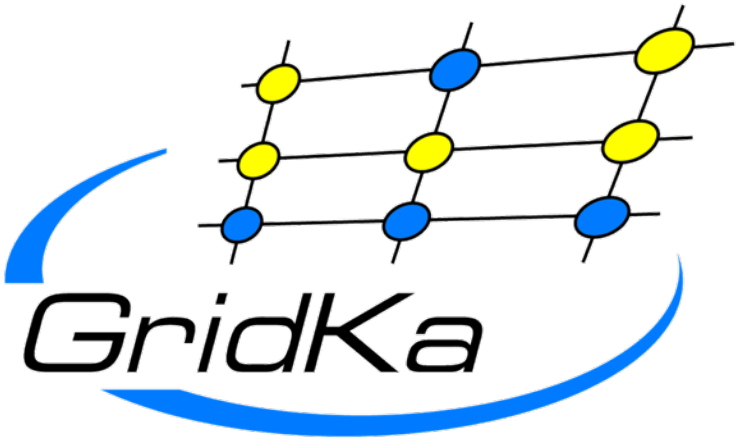
- Process and network traffic monitoring tool
  - Monitoring process trees of batch system jobs
  - Network monitoring based on libpcap
  - Logging of network traffic on packet level
  - Association of process and network data
- Internal preprocessing
  - Categorisation into internal and external network traffic
  - Grouping of traffic by connection

- Stable operation for several months
- Already proven to be useful
  - For example: Detected a class of misbehaving jobs producing loads of external traffic because of a software bug
    - Firewall was heavily loaded and GridKa not reachable
    - Due to fine-grained monitoring, fast identification and reaction on issue

Operation on GridKa Cluster

- Tier 1 computing centre
  - Supports all four major CERN LHC collaborations as well as AUGER, BELLE2, COMPASS, and others
  - Operation of ~600 worker nodes
  - Provision of ~13,000 job slots
  - Batch system: Univa Grid Engine

- Prototyp monitoring on 2 racks
  - 2x16 worker nodes, ~800 job slots
  - Stable operation since July ‘14

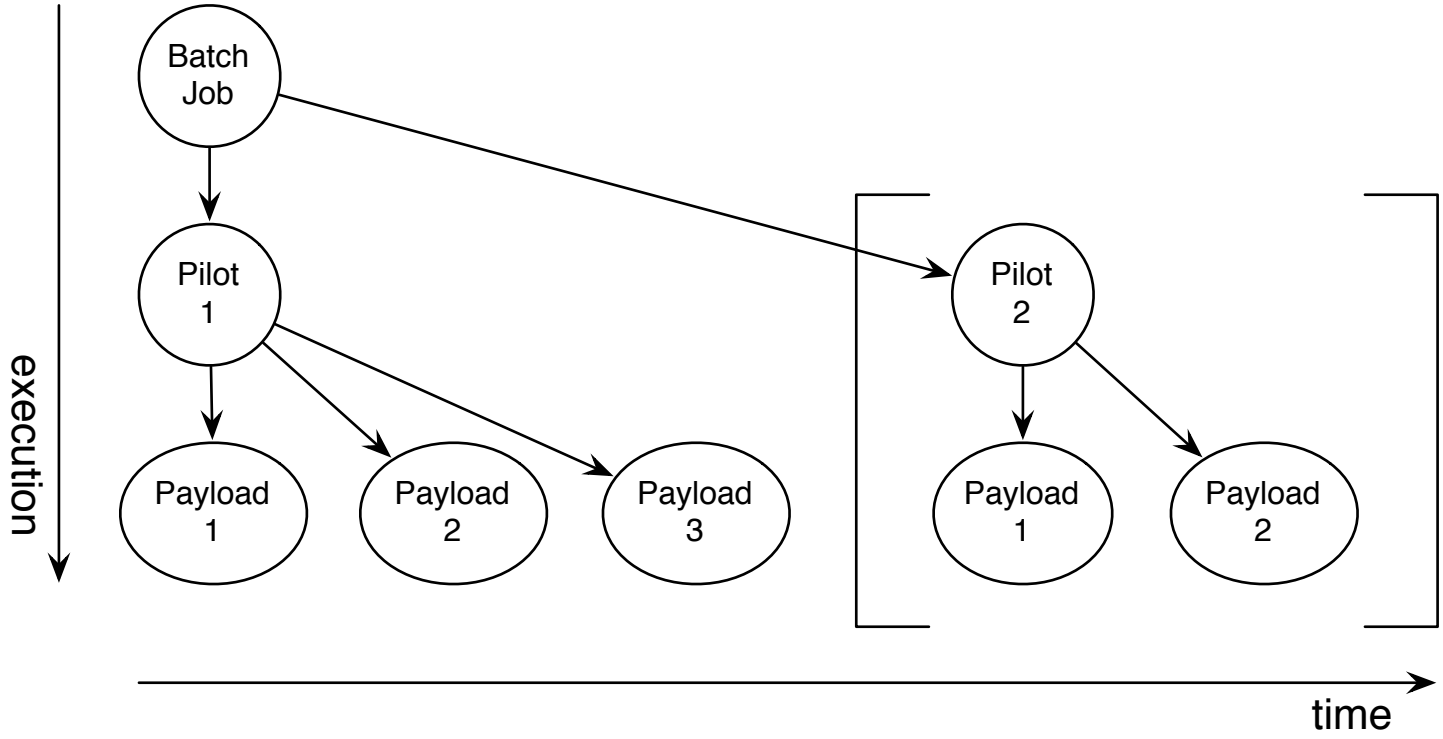


Demonstrative Monitoring Sample

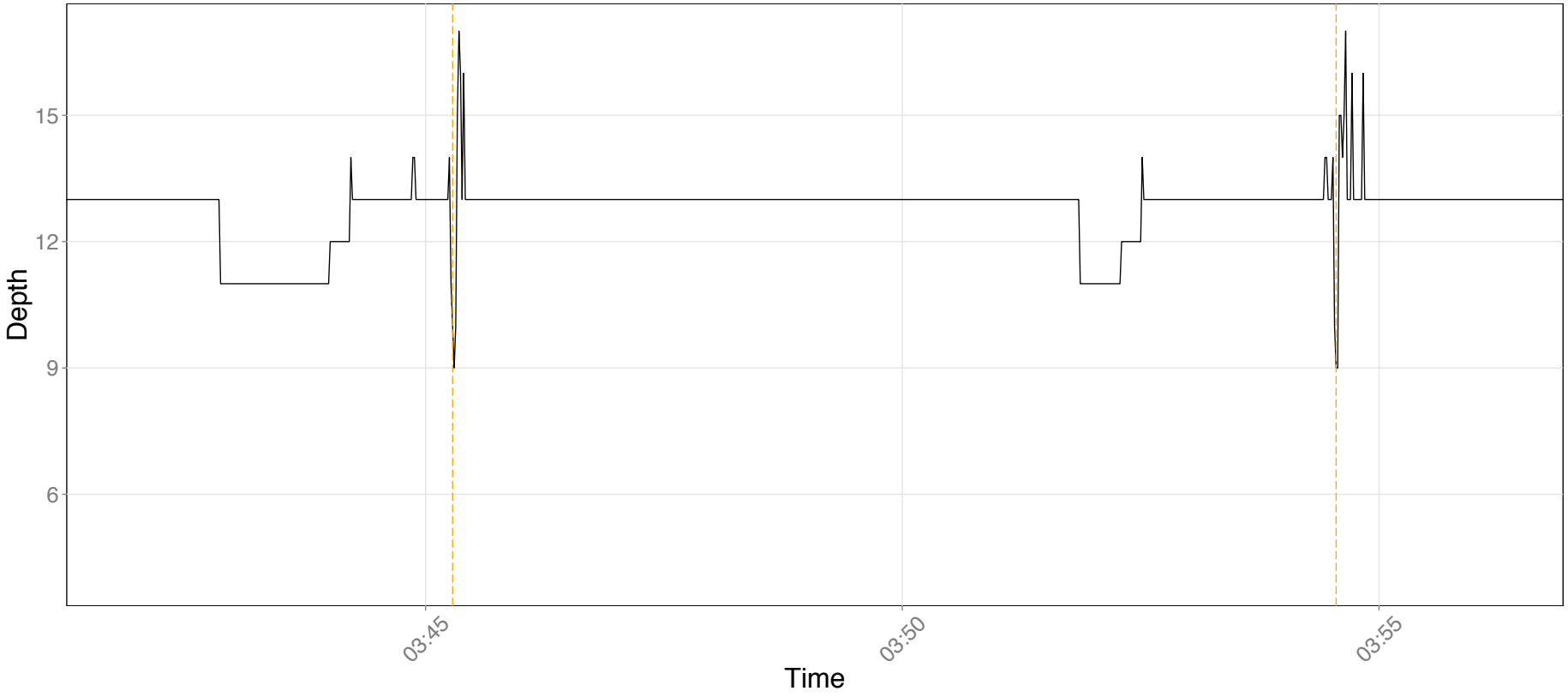
- Monitoring data subset for exploratory data analysis
  - 15 day timeframe
  - 4 worker nodes, ~100 job slots
  - ~10.000 batch jobs
- Complete UNIX process tree for each batch job
  - Batch system processes, job wrappers, watchdogs, …
  - Wide range of complexity (234 - 3,983,680 sub-processes)
- Detailed process information
  - Name, cmd, $?, …
  - Per connection source/destination IP, port, data rate, packet count, …
- Selection of CMS pilots by
  - CMS Pool Account ownership
  - HTCondor (glidein)

Payload Splitting Approach

- Payloads started and managed by one parent process
  - Cut on pilot process children
- Sequential management/execution of payloads
- Internal/external communication mainly takes place in payloads

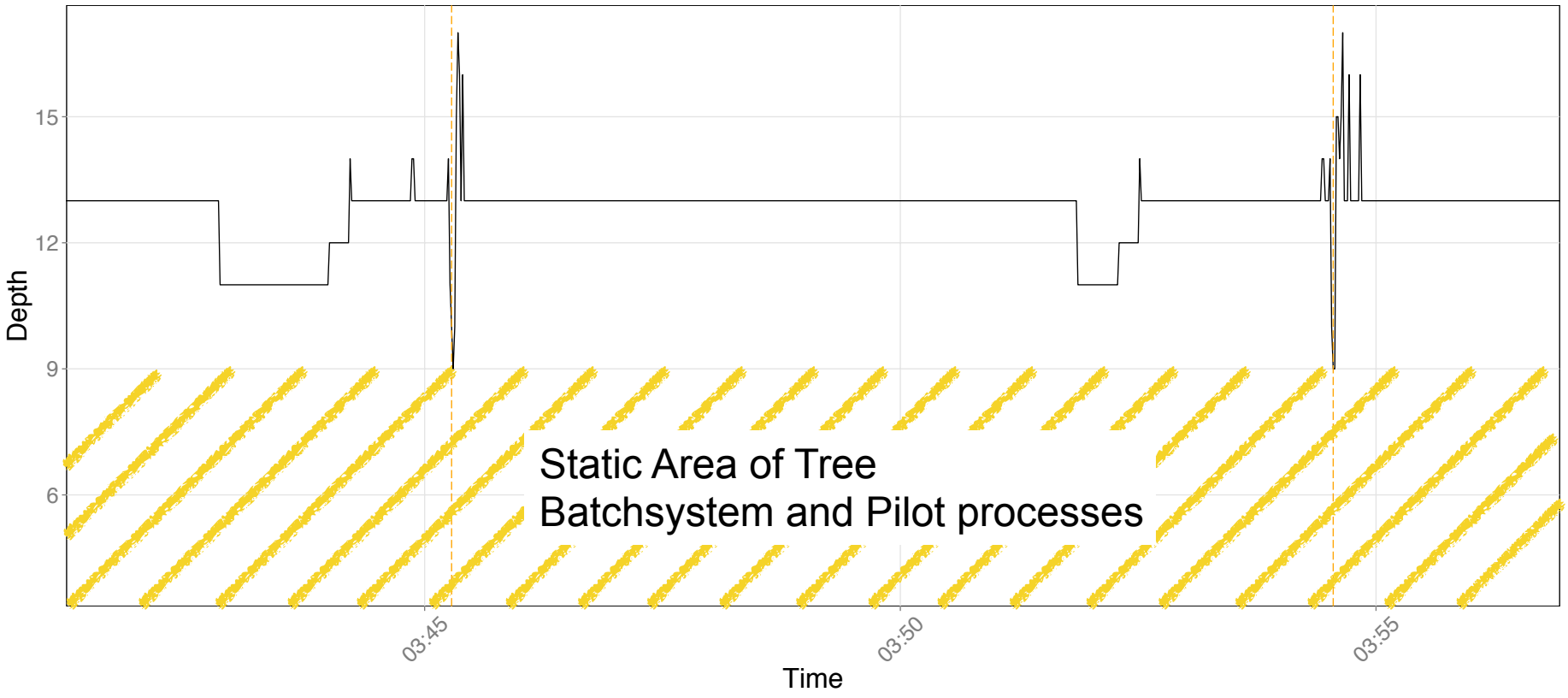


Payload Splitting Based on Tree Depth



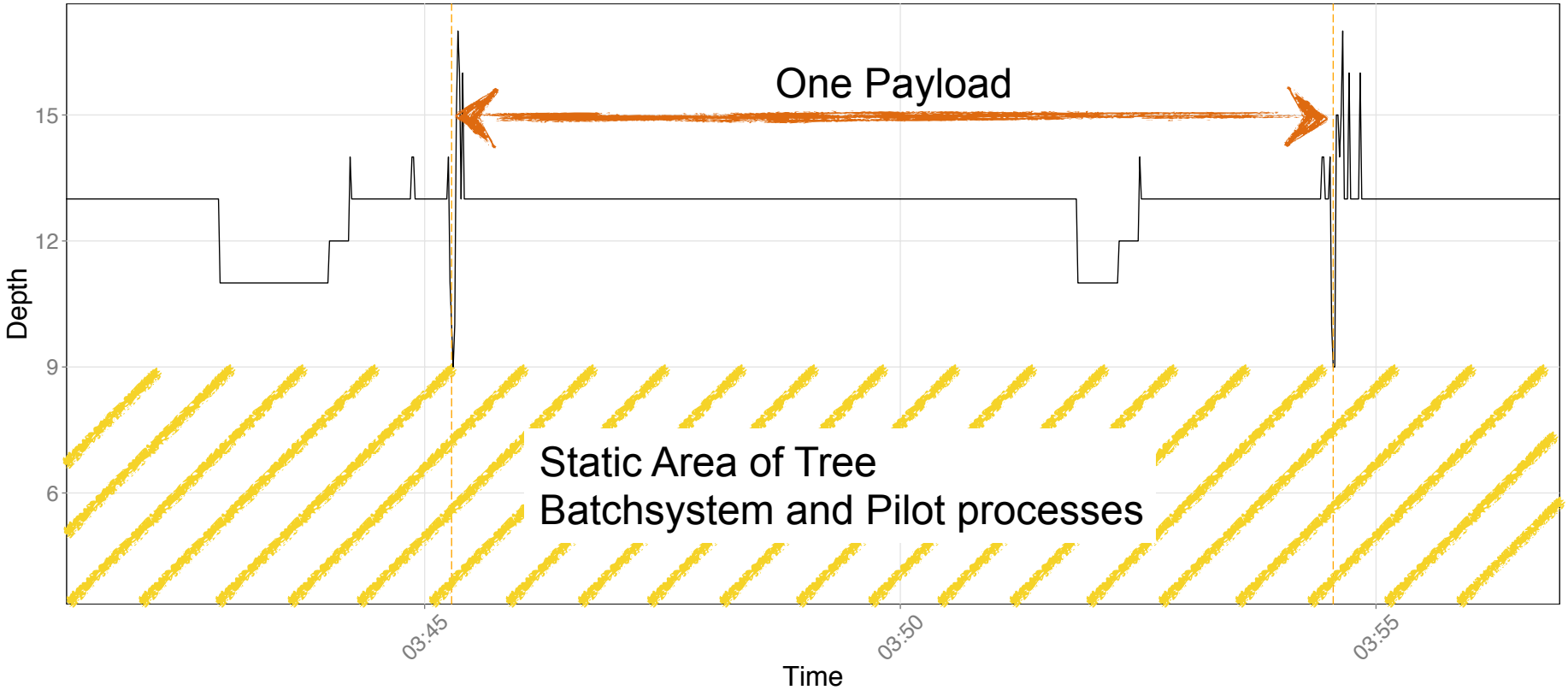
- Prototype for generalised payload recognition based on tree depth
- Efficient on-line estimation of payload domain and activity

Payload Splitting Based on Tree Depth



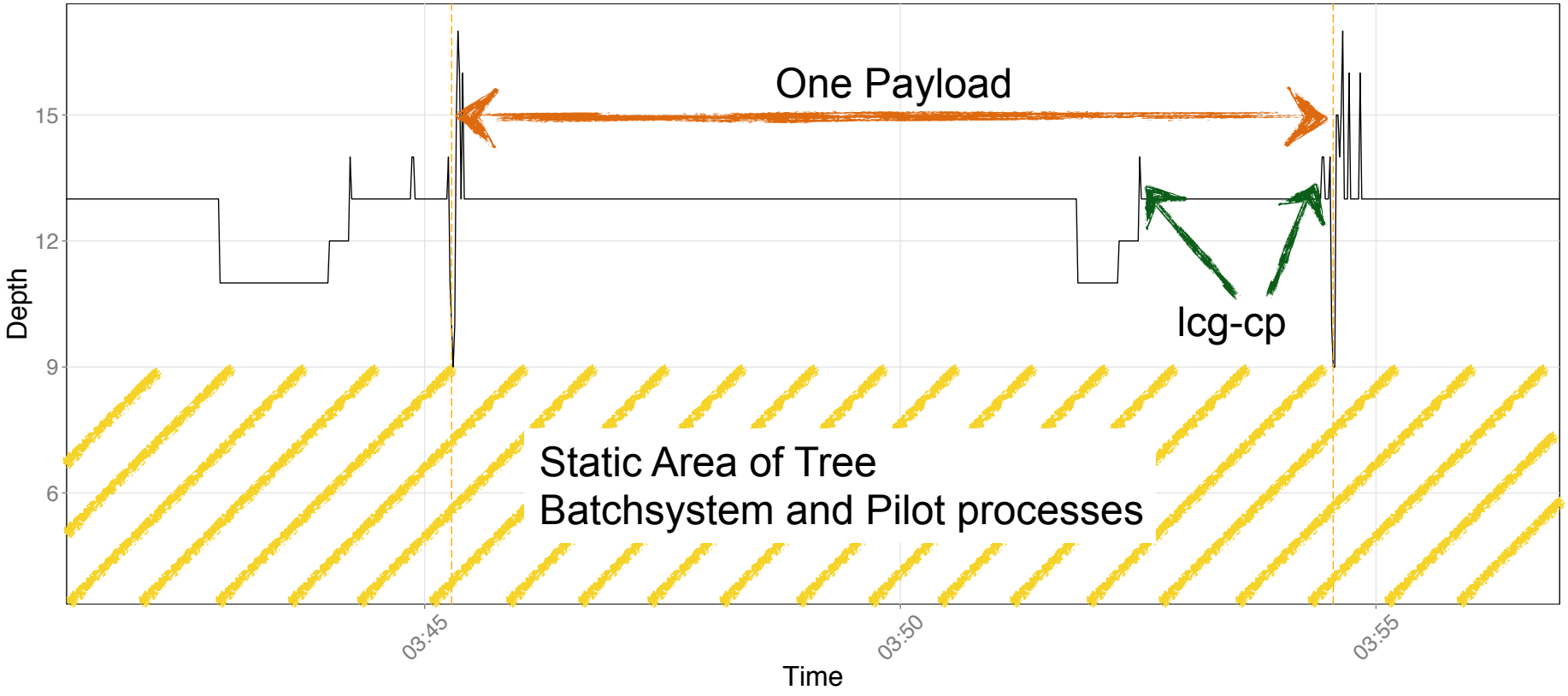
- Prototype for generalised payload recognition based on tree depth
- Efficient on-line estimation of payload domain and activity

Payload Splitting Based on Tree Depth



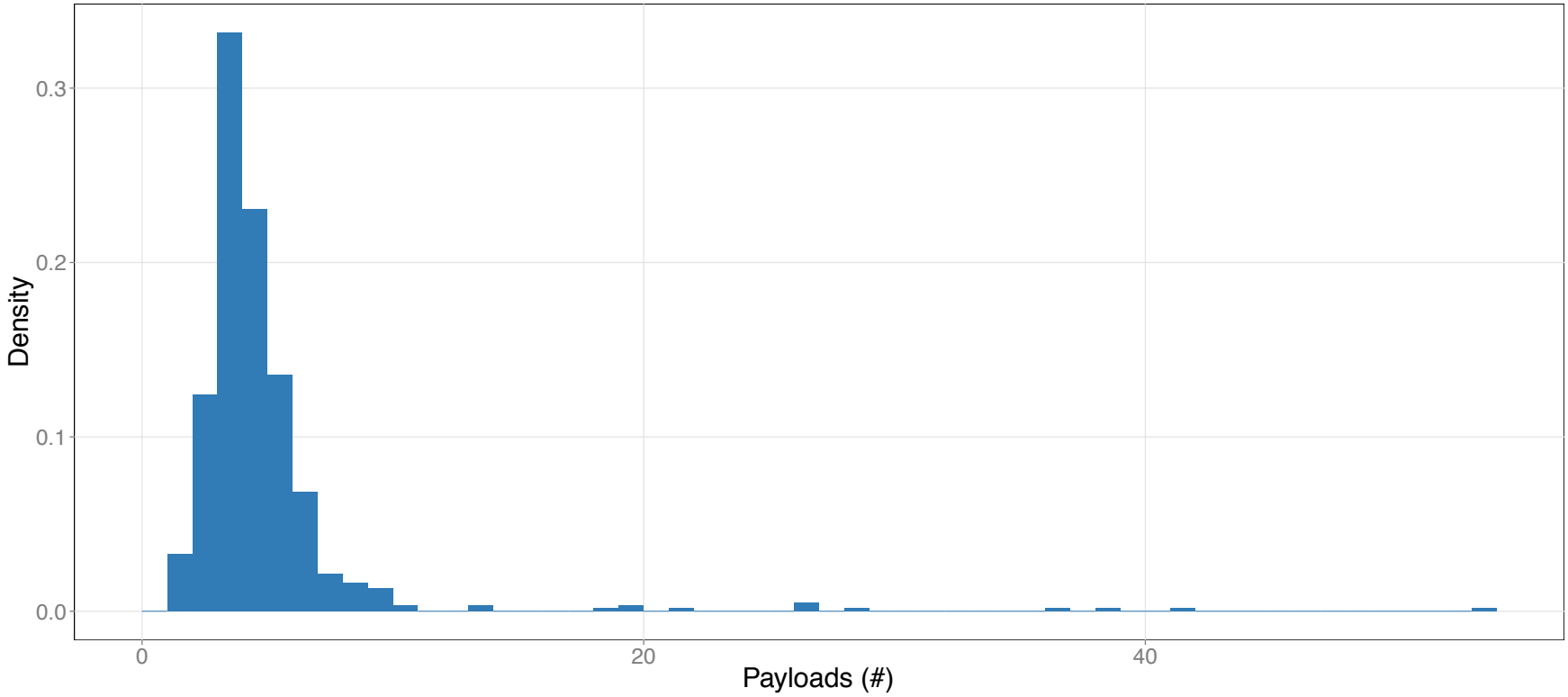
- Prototype for generalised payload recognition based on tree depth
- Efficient on-line estimation of payload domain and activity

Payload Splitting Based on Tree Depth



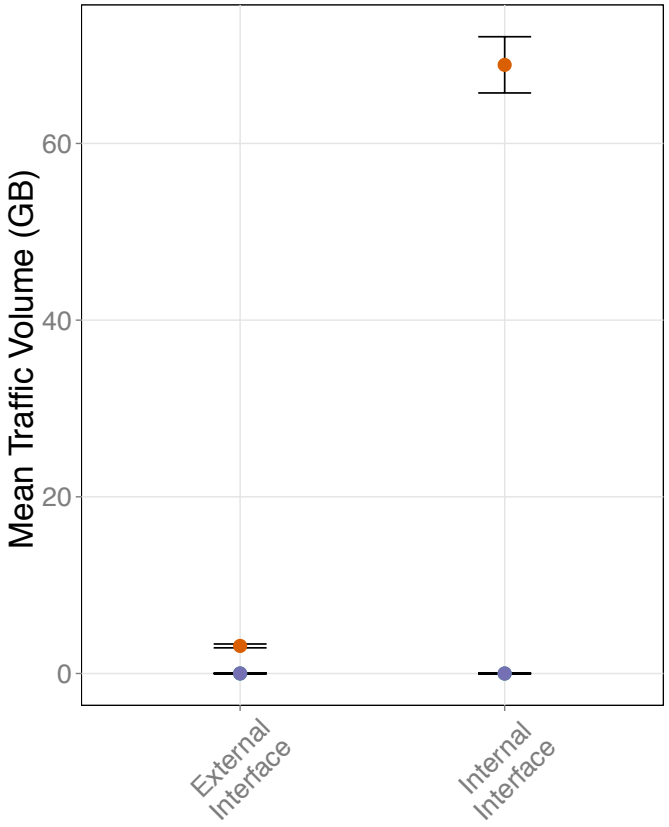
- Prototype for generalised payload recognition based on tree depth
- Efficient on-line estimation of payload domain and activity

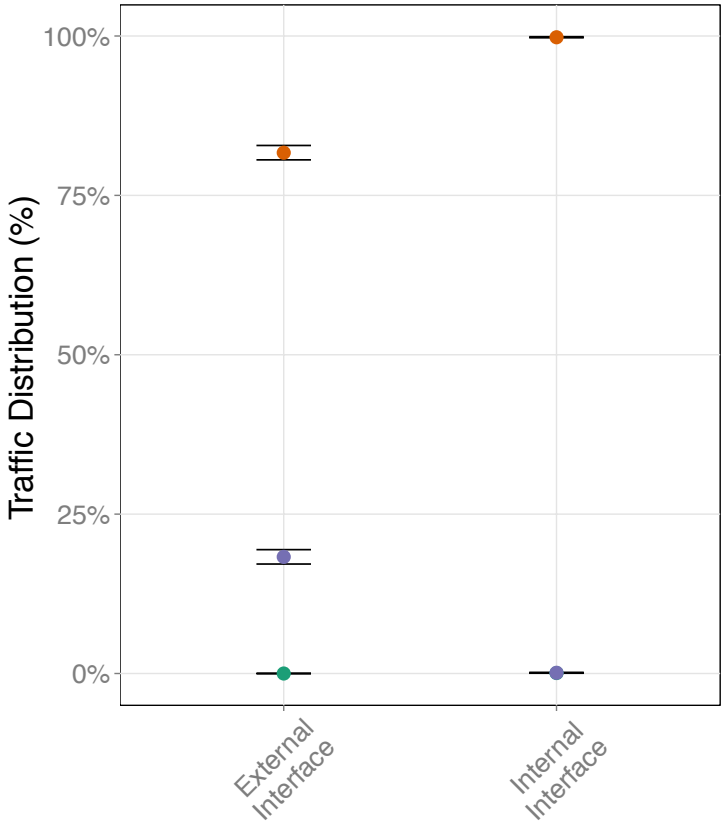
Payloads in CMS Pilots

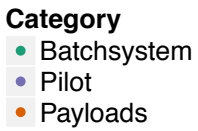


- Pilots always combine multitude of payloads
- Payload count varies widely between pilots

Traffic Distribution by Component

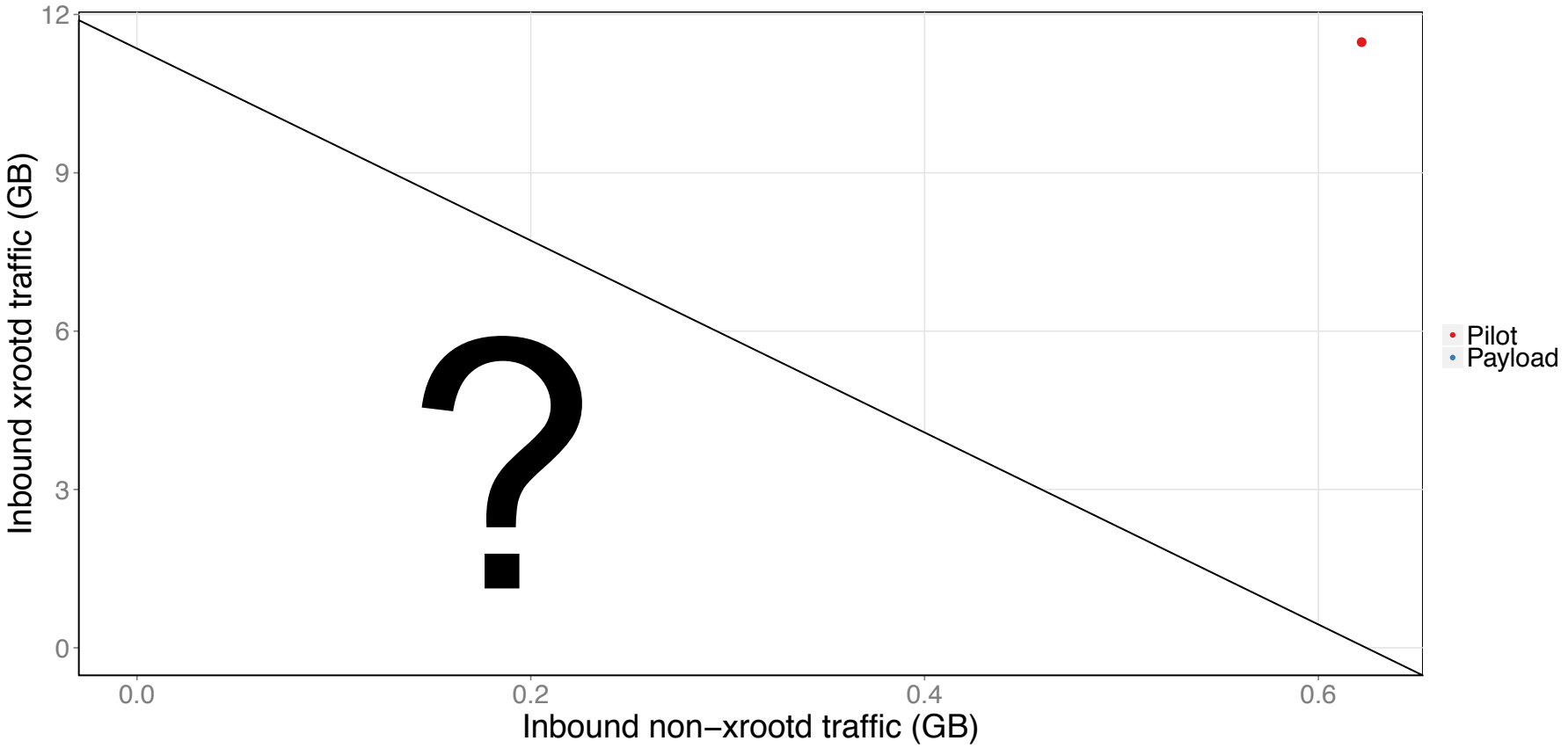




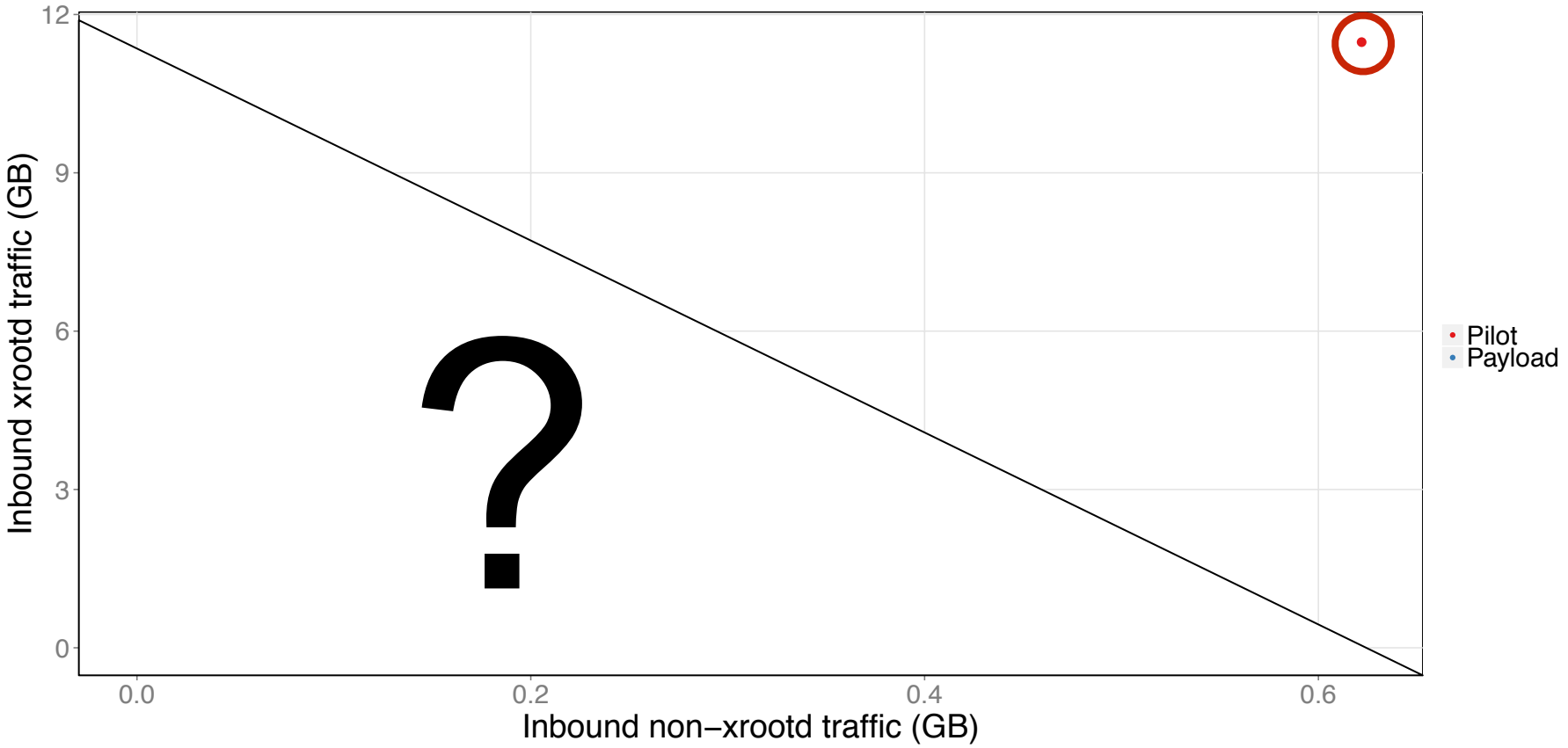


- Traffic generation dominated by payload processes
- Traffic footprint clear indicator for payload identification

Pilot monitoring vs. payload monitoring

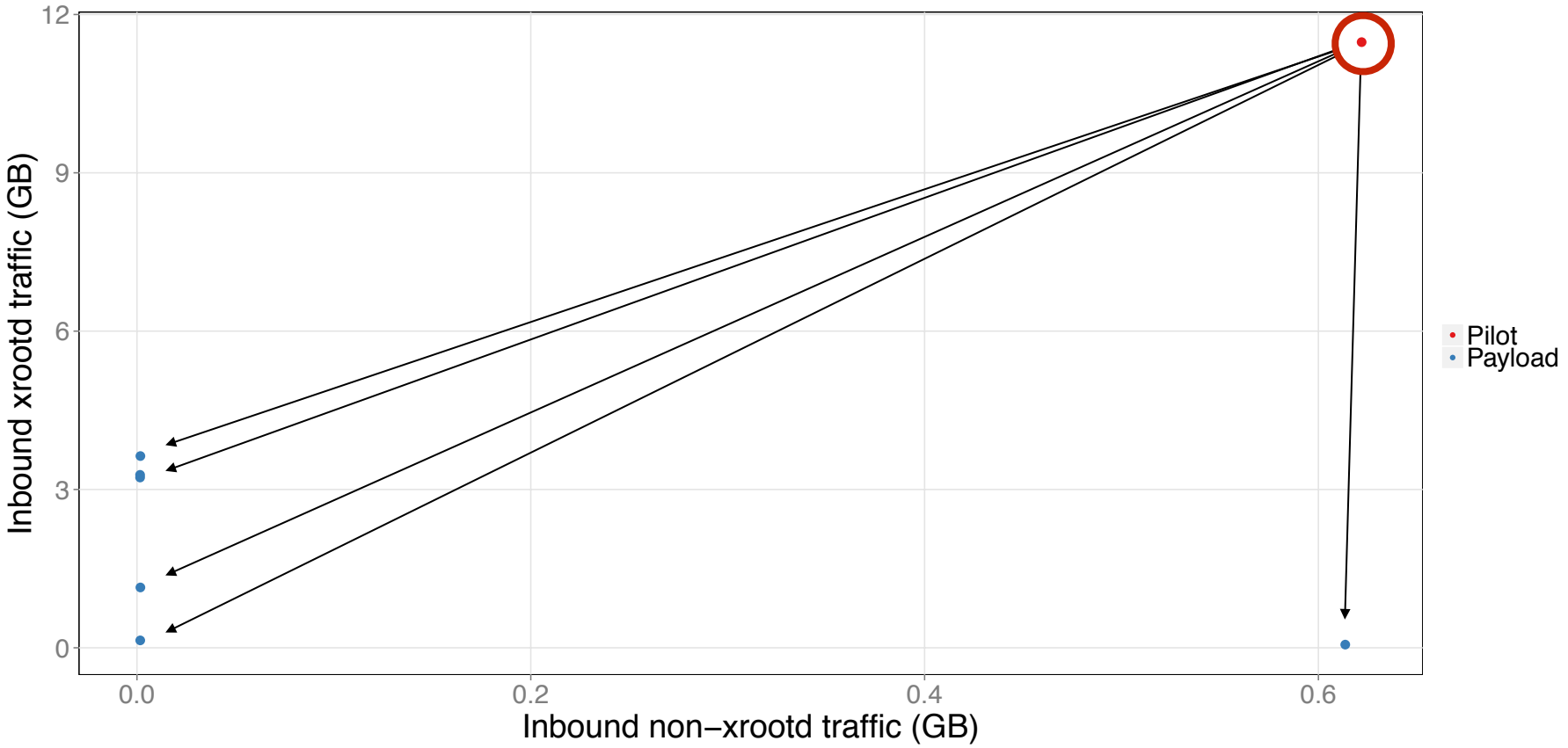


Pilot monitoring vs. payload monitoring



- Pilot merges payload features into single measurement

Pilot monitoring vs. payload monitoring



- Pilot merges payload features into single measurement
- Payload splitting reveals individual resource consumption

Summary and Outlook

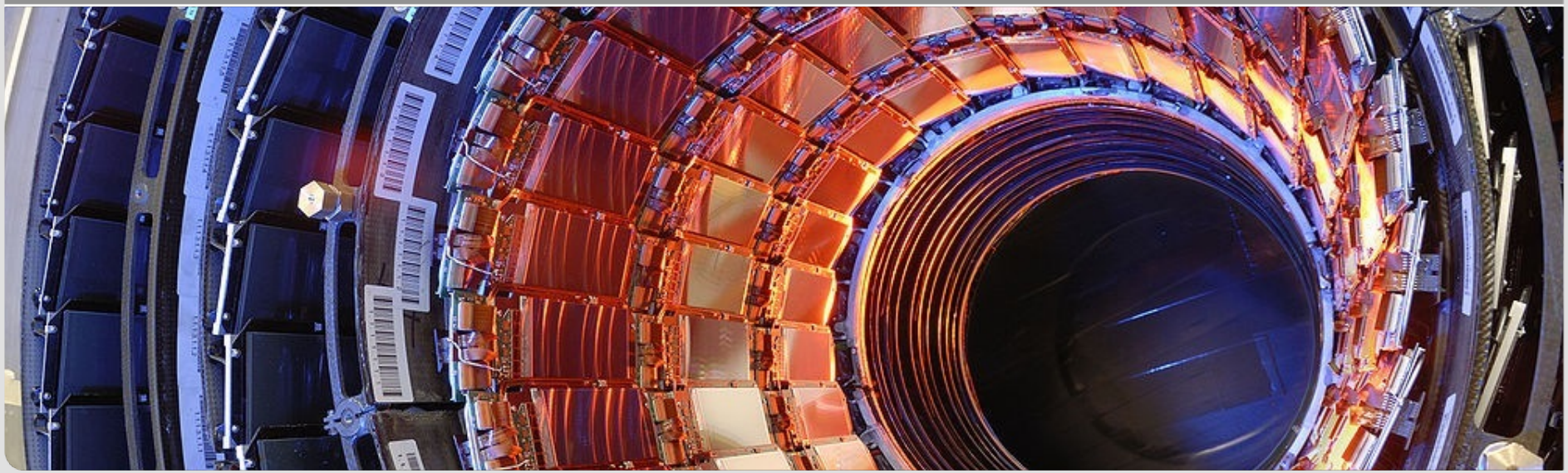
- Fine-grained job monitoring tool
  - Process tree recording
  - Per process connection tracking
- Tracking of payloads in executing pilots
  - Independent of VO monitoring
  - Reestablished “job”-level monitoring
  - Detailed per job traffic monitoring

- Extension of payload identification
  - Generalisation of pilot identification beyond CMS
  - Near realtime recognition
- Basis for on-line assessment of payloads
  - Anomaly detection and reaction
  - Identification of payload types
  - Evolution of usage patterns

Karlsruhe Institute of Technology

Questions?

Steinbuch Centre for Computing, GridKa



Payload Monitoring Tool