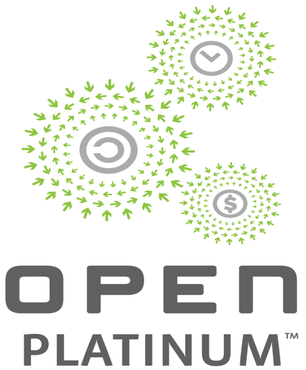

Open. Together.
OCP
SUMMIT

Exploiting the true potential of At-Scale debug

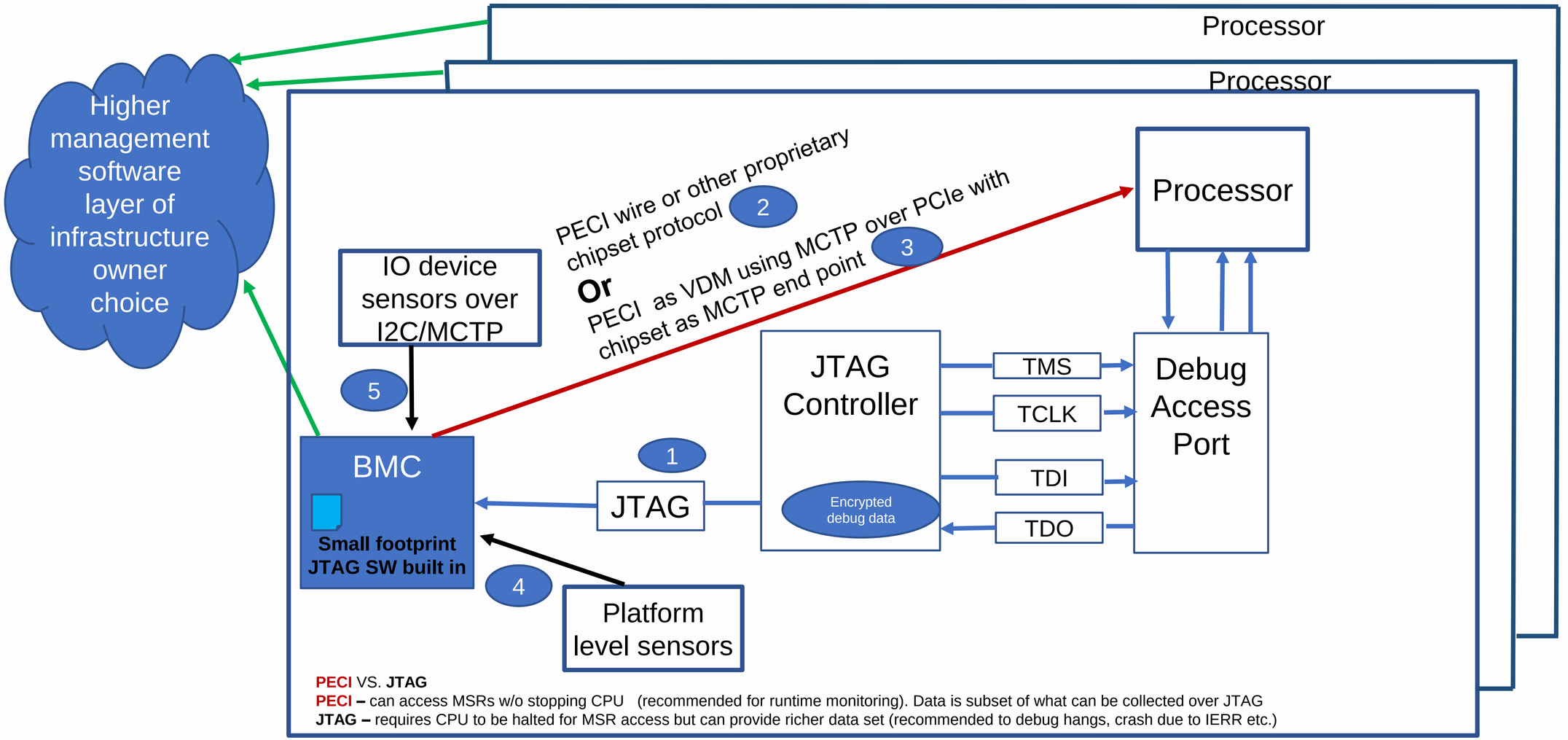
Sumeet Kochar, Lenovo
Theodros Yigzaw, Intel
Gundrala Devender Goud, MSFT



The Only Good Downtime is NO Downtime

- Cost of downtime is big in terms of $ but it is not just $
- Downtime analysis should also consider
  - “gray failures” – infrastructure up but not performing as expected
- Right data = “telemetry” from the systems is key to weeding out such issues before they persist and become a problem at scale
- Historical data is as important as data at the point to failure
- This presentation will dive into a reference implementation for server debug at scale – ***much more than JTAG connection to host chipset from BMC to mimic an In Target Probe at scale***

Platform hooks for Debug at Scale ?



PECI VS. **JTAG**
PECI – can access MSRs w/o stopping CPU (recommended for runtime monitoring). Data is subset of what can be collected over JTAG
JTAG – requires CPU to be halted for MSR access but can provide richer data set (recommended to debug hangs, crash due to IERR etc.)

OPEN SYSTEMS FIRMWARE

Reference Architecture

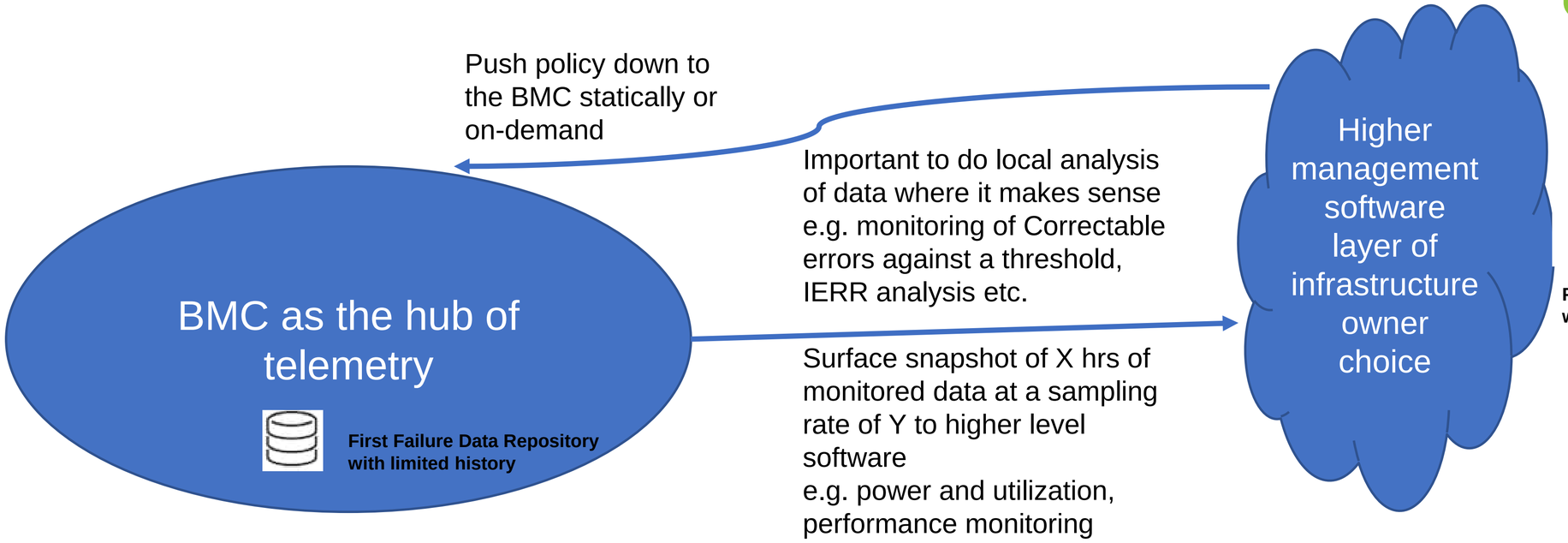
Target Capabilities

- Failure Prediction
  - Access to telemetry data (e.g. memory rd/wr bandwidth, cache allocation, power control events …etc.) for predicting Gray Failures
  - Predicting failures based on environmental conditions (high temperature, voltage …etc.)

- Failure Diagnosis
  - Some failures are hard to reproduce in a lab environment
  - Happen only at-scale in a given context, running a given workload in a given environment
  - Collecting failure and telemetry data at the point of failure is essential to diagnosing them faster and more accurately
  - Encrypted extraction of Intel micro-architectural state is key for certain hard-to-diagnose failures.

Data Analysis at the edge or at higher level ?

OPEN SYSTEMS FIRMWARE



Reference Architecture

Types of telemetry possible

OPEN SYSTEMS FIRMWARE

Trained algorithms for detecting CPU access anomalies

BMC as the hub of telemetry

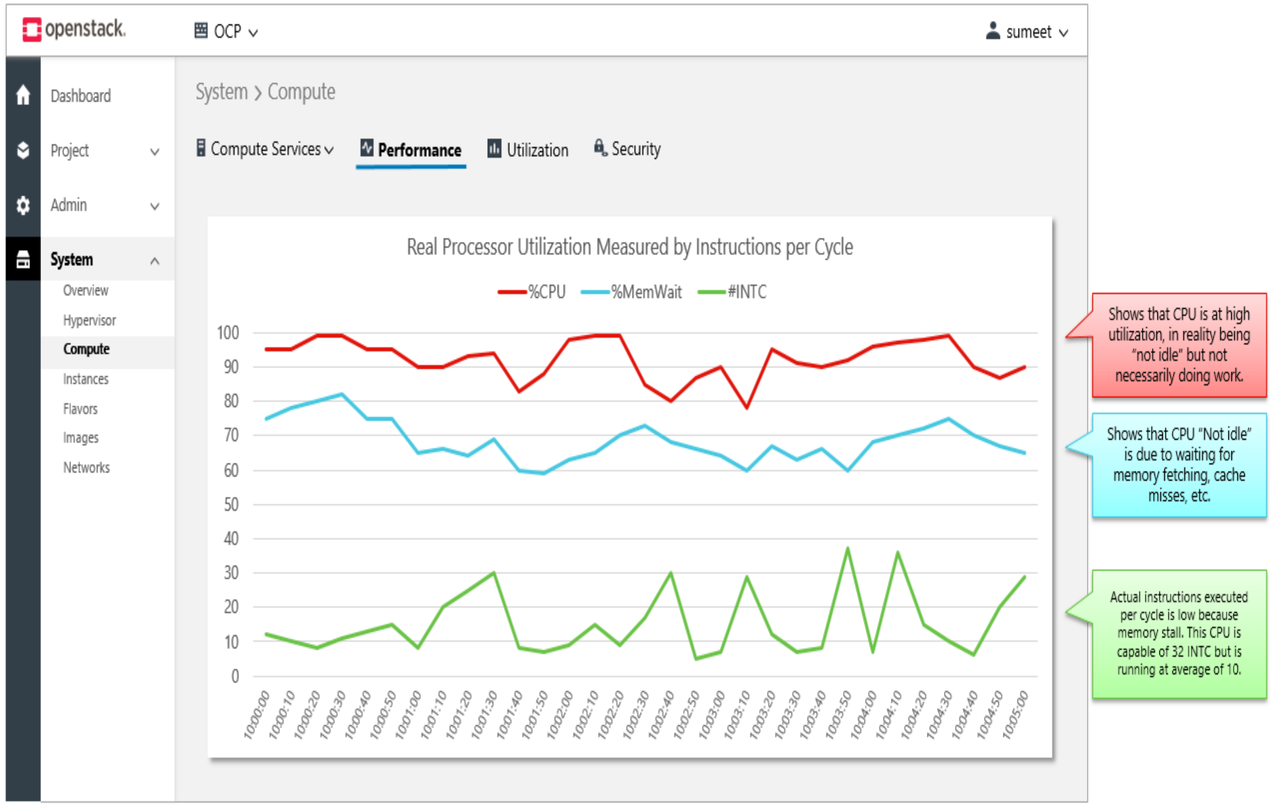
Utilization and power consumption

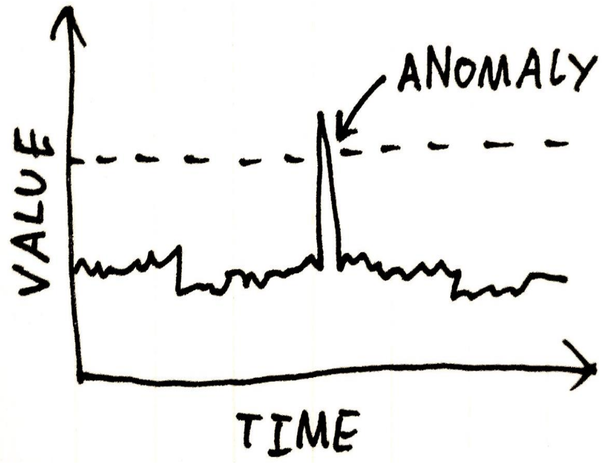
Debug at Scale

- **“Gray Failures”**
- **Crash dumps**
- **Perf**

Reference Architecture

Visual mapping of telemetry w/ plugins





OPEN SYSTEMS FIRMWARE

Reference Architecture

ASD – Cloud Use Models

- Runtime Monitoring – Reliability & Availability
  - Secured Telemetry – non-intrusively pull heart-beat data from remote nodes
  - Perform Predictive Analysis and take proactive actions to migrate customers' workloads to new machines
  - Compute Workloads: Multi-tenant Stateless computing - Availability is key
  - Storage Workloads: Stateful computing/pass through – Data Retention, Protection and Storage – Reliability is key
- Downtime Analysis - Serviceability and Diagnosablity
  - Remote OOB Debug interfaces required
  - FRU identification for expedited root-cause and part replacement tools/interfaces
  - Multi-tenant serviceability could be a challenge

Call to Action

- Provide feedback on the high level idea – BMC centric vs. host centric
- Welcome thoughts on type of telemetry that is of importance
- We plan to start contributing this framework to OpenBMC by end of this year. Help review and extend capabilities when available.


Open. Together.
OCP Global Summit | March 14–15, 2019
OCP
SUMMIT