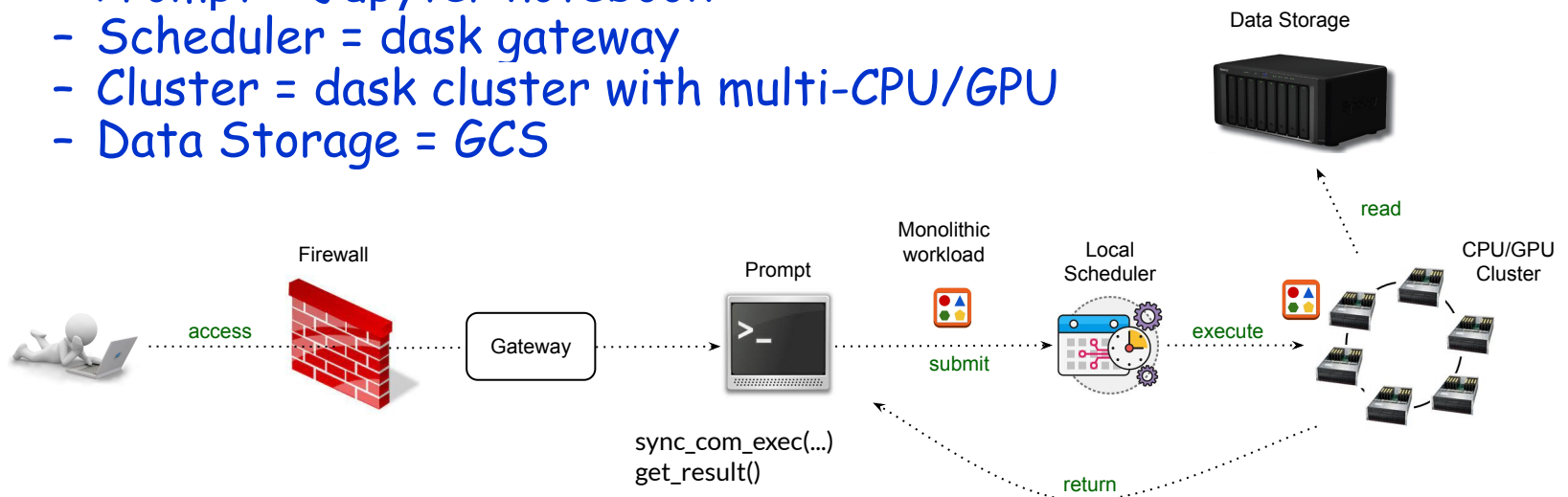


Analysis Tools and New Workflows

Tadashi Maeno (BNL)

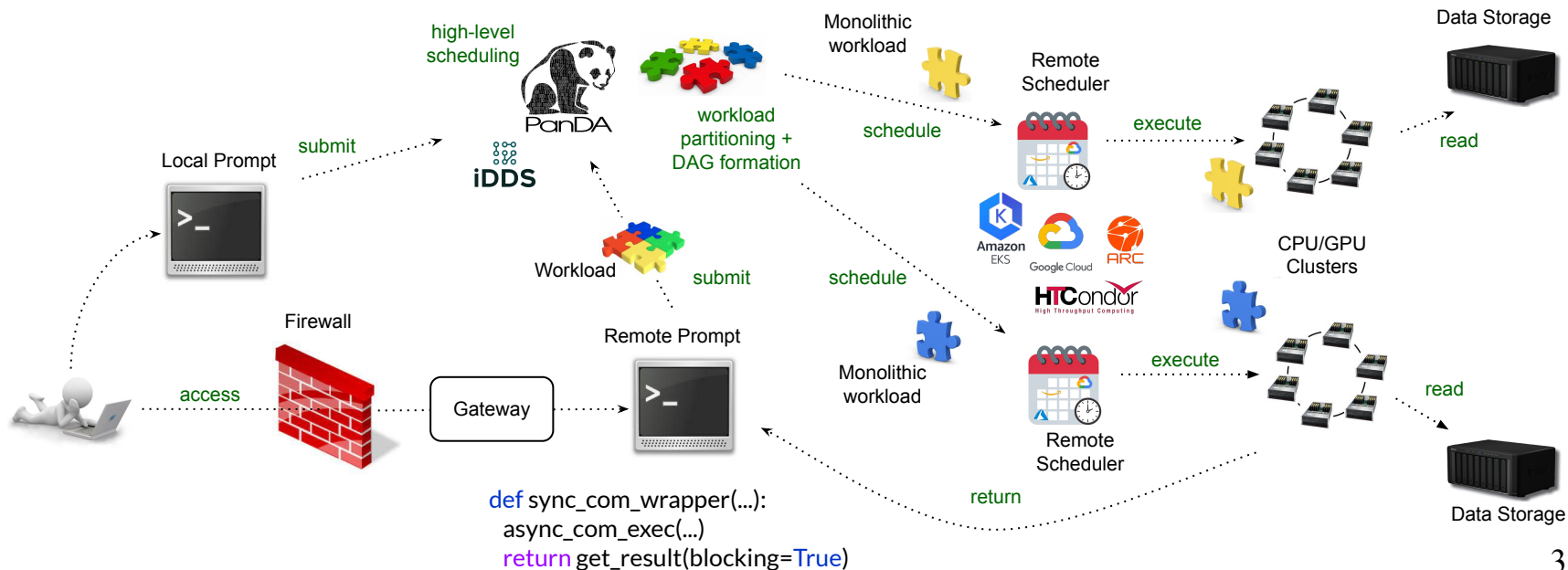
Ordinary Interactive Analysis with “Distributed” Resources

- In many SW frameworks
 - “Distributed” = “multi-node”
 - Computing resources in LAN managed by a local scheduler
 - User’s remote access via gateway service
 - Interactivity = command prompt + synchronous command execution + short latency
 - Reading of pre-placed data from local storage or on-demand data reading from remote storage through local cache, with (a)synchronous data ingestion
 - “Monolithic” workload with frequent interactions among CPU/GPU nodes for AllReduce etc
- E.g. dask@Google
 - Gateway = Jupyter hub
 - Prompt = Jupyter notebook
 - Scheduler = dask gateway
 - Cluster = dask cluster with multi-CPU/GPU
 - Data Storage = GCS

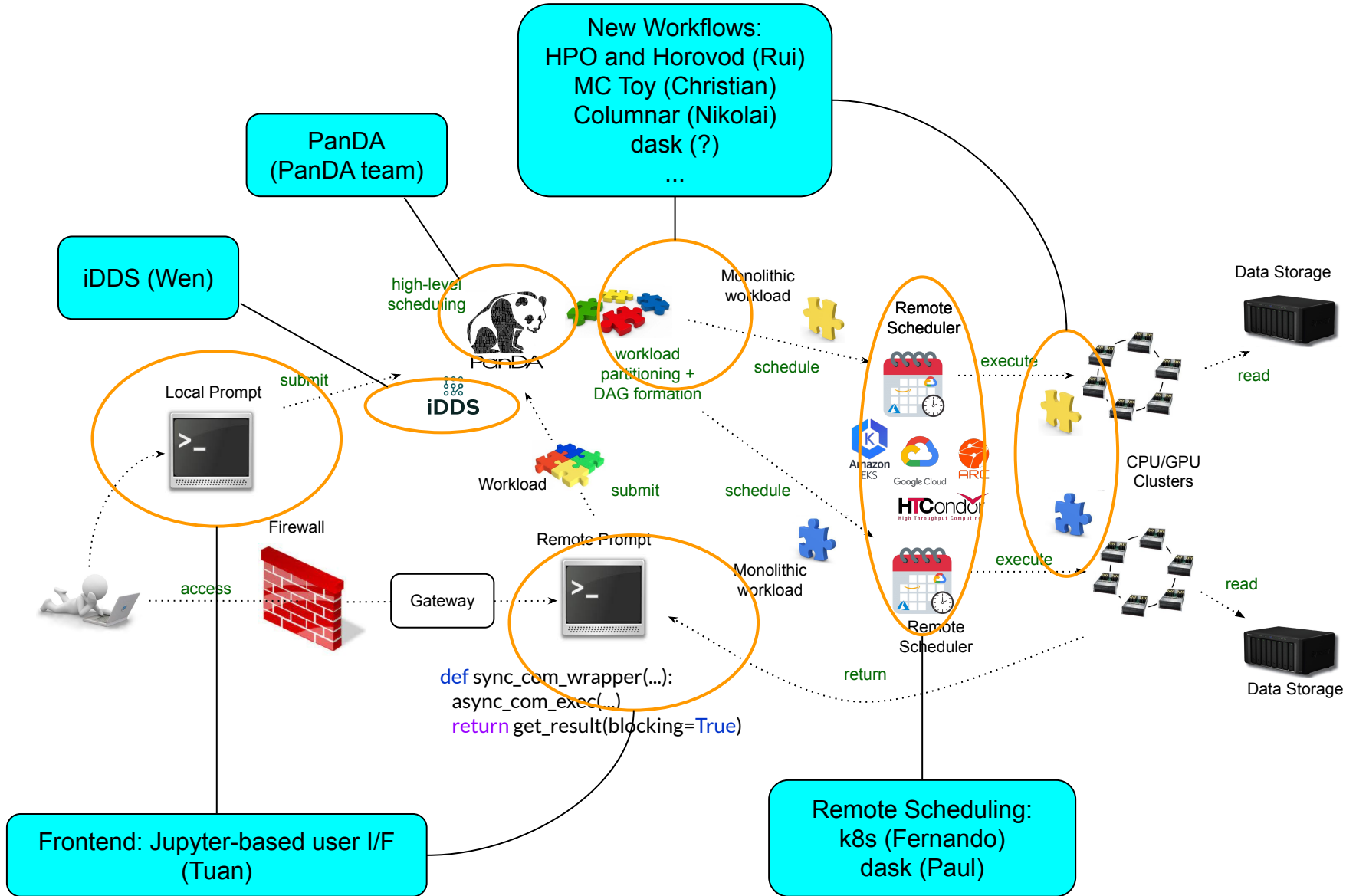


Pseudo Interactive Analysis with ATLAS Distributed Resources

- In ATLAS, "distributed" = "geographically distributed"
- Pseudo interactive = command prompt + synchronous command wrapper + asynchronous command execution + "acceptable" latency
- Central scheduler = PanDA + iDDS
- Local or remote command prompt
- Workload partitioning and processing with DAG
- Remote scheduler = CE, k8s scheduler, harvester@edge, ...
- Any kind of analysis (columnar, declarative, MPI, ...) = payload from system's point of view + user I/F for each analysis



Pseudo Interactive Analysis with Distributed Resources



Latency

- **Acceptable latency TBD**
 - Depending on data volume, intensity of computation, and user's expectation
 - Tens of minutes, several hours, a couple of days, ...
- **Major contributions**
 - **Task and job creation time**
 - Dedicated JEDI and Panda server nodes
 - Express share for analysis
 - Shorter daemon cycles
 - **Queueing time in remote schedulers**
 - Dedicated computing resources, preemption, on-demand cluster spin-up, ...
 - **Tail in task completion time**
 - **Stage-in/data-ingestion and stage-out time**

Conclusions

- The system is absorbing new workflows
 - Any data processing/ingestion tool, mechanism, framework as a backend or payload
- Leveraging the capability to integrate geographically distributed resources
 - Further scale-up with more computing resources across various resource providers
- Workload partitioning and processing with DAG are crucial
 - Parallel execution
 - Complicated flow
- Ideally, transition from interactive to pseudo interactive should be transparent from user's point of view
 - Latency
 - User interface