barkhauseninstitut.org

barkhausen institut

Slashing the Disaggregation Tax in Heterogeneous Data Centers with FractOS

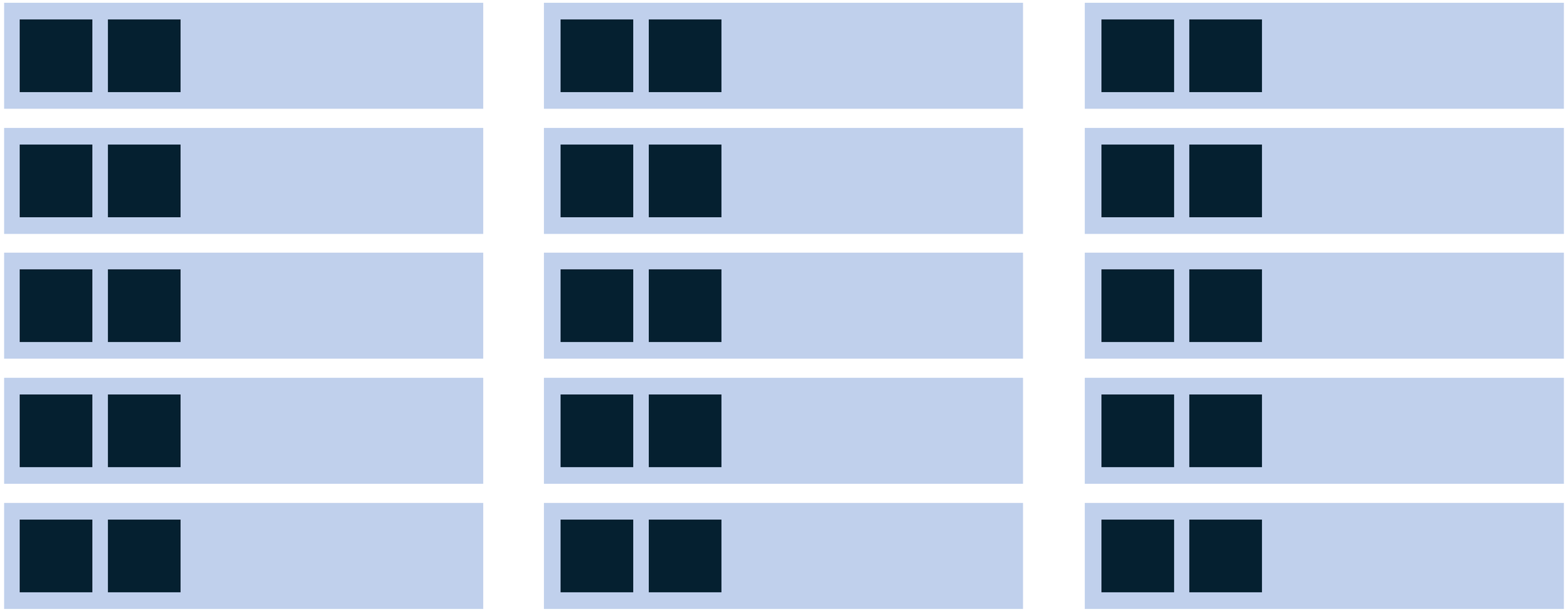
Lluís Vilanova, Lina Maudlej, Shai Bergman, Till Miemietz, Matthias Hille, Nils Asmussen, Michael Roitzsch, Hermann Härtig, Mark Silberstein

Context & Contribution

Implement a decentralized application substrate for disaggregated data centers

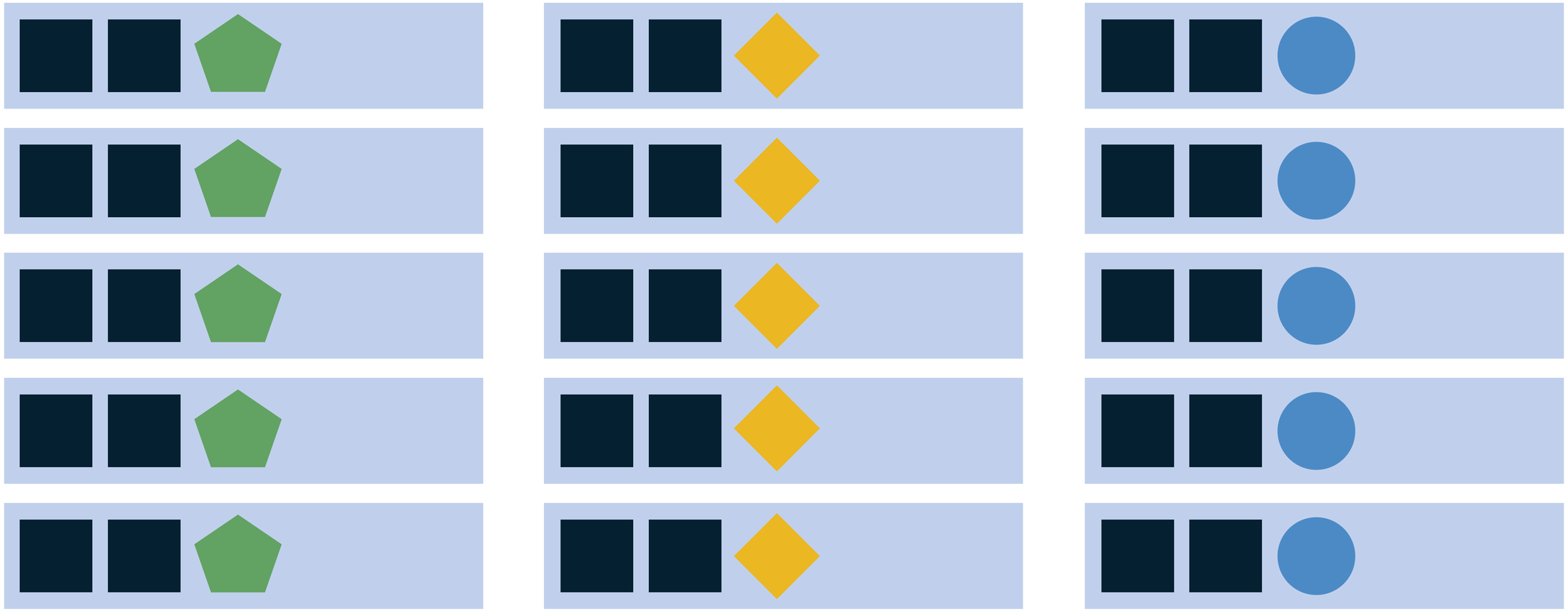
- distributed capability system
- continuation-based invocation
- isolated OS layer

Data Center Hardware



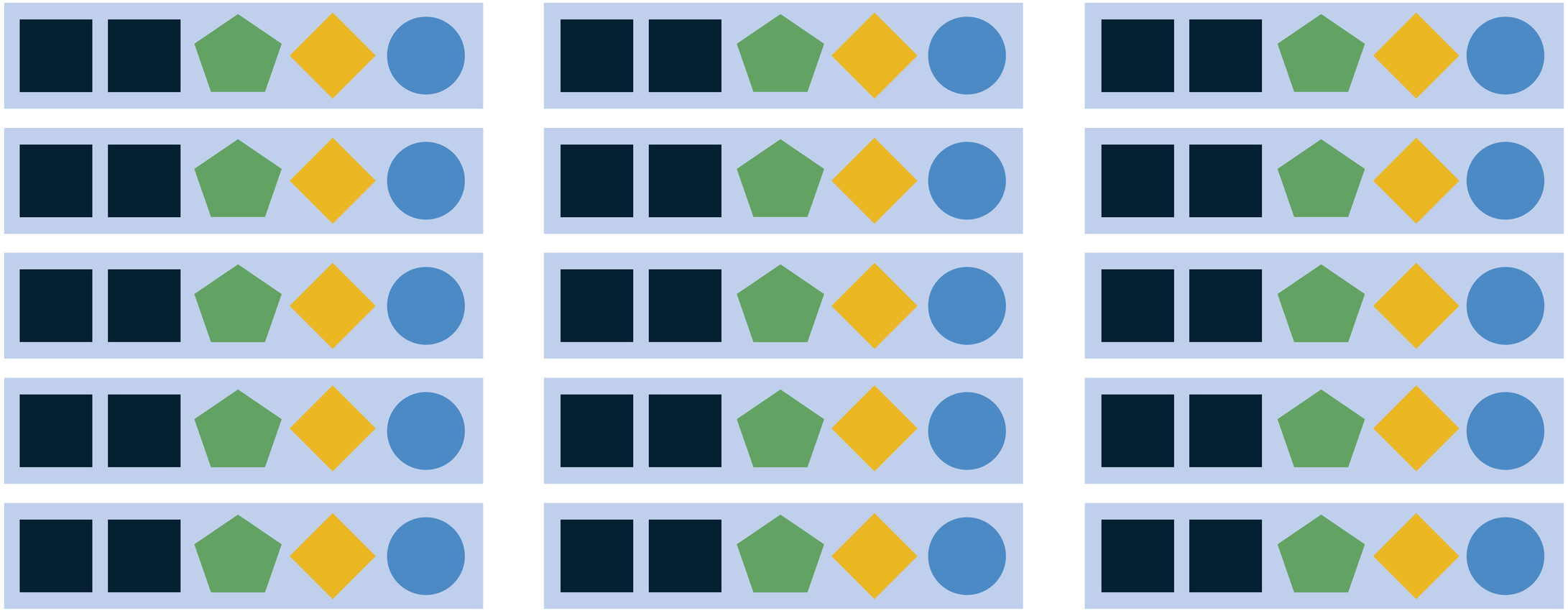
Standard processors: good for everything, but not energy efficient!

Data Center Hardware: Specialization



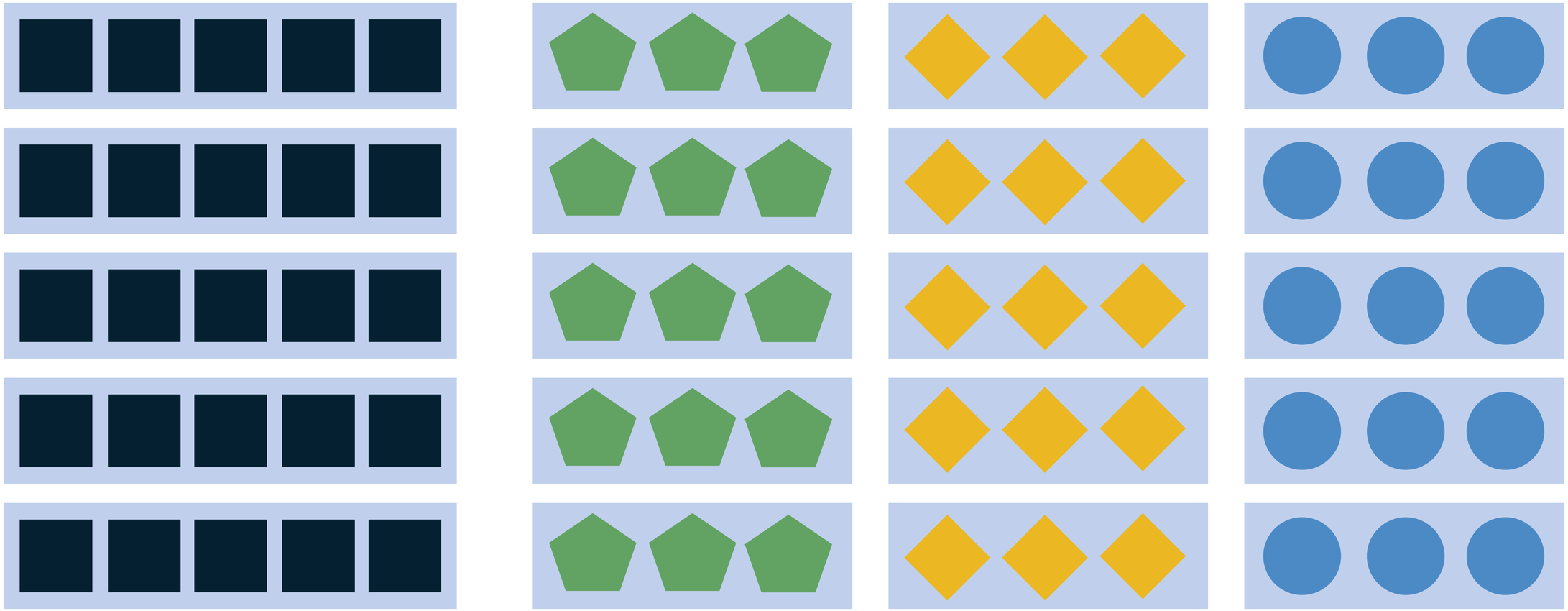
Specialized processors for specific tasks: efficient, but we lose flexibility!

Data Center Hardware: Specialization



Specialized processors everywhere: more than needed, wasteful!

Disaggregated Data Center



Pools of equal resources plus a fast network to combine them.

Disaggregated Data Center

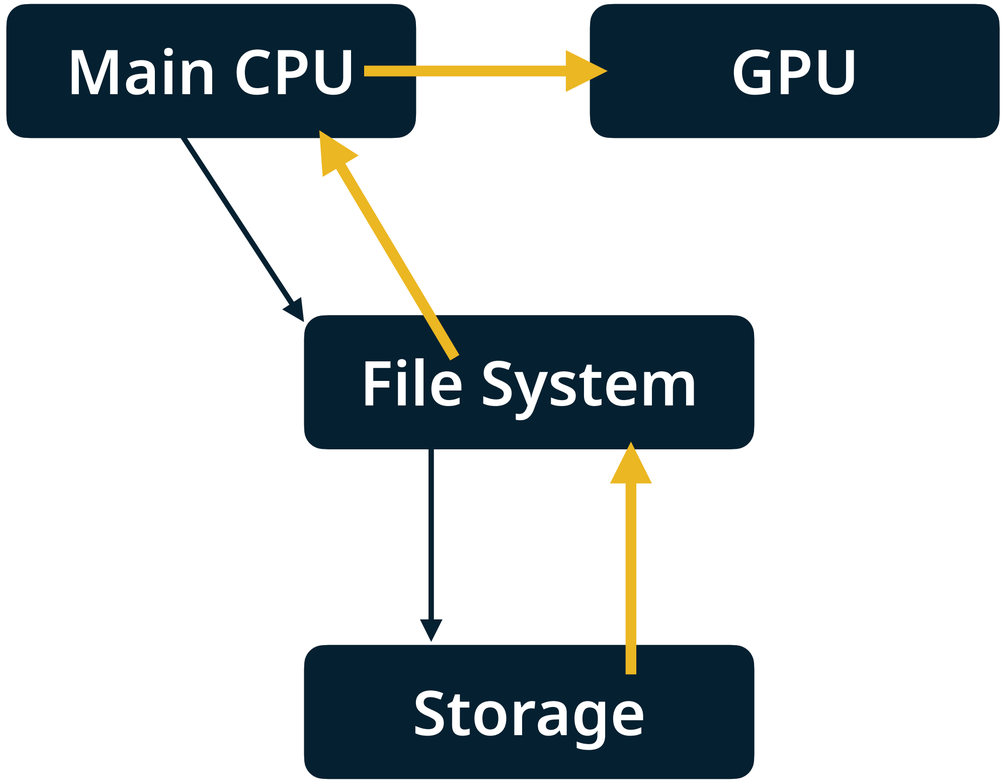
Cables are Slow!

3 GHz processor speed

- a single computation takes 0.3 nanoseconds
- light travels 9 centimeters in 0.3 nanoseconds
- a cable of 90 meters is 1000 times longer!
- operations involving different processors slowed down by 1000×

We must avoid network interactions!

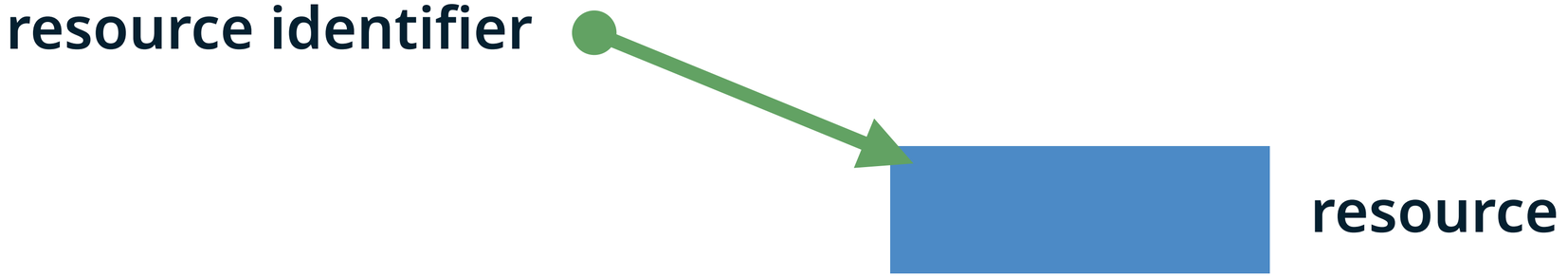
Data Transfer Between Services



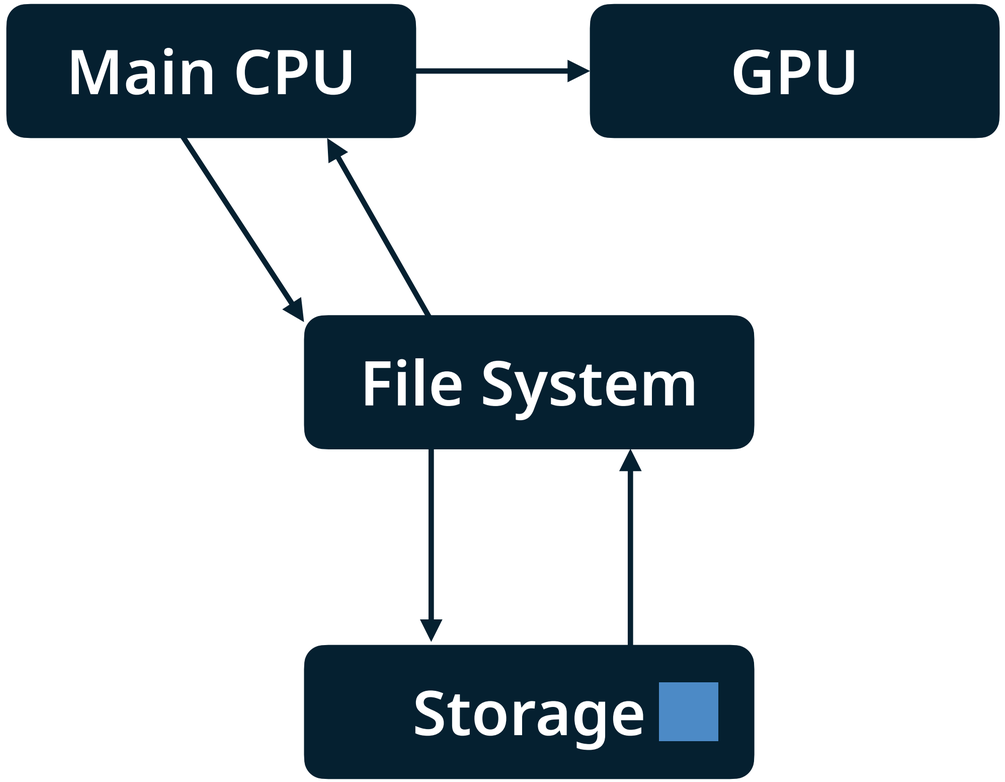
How can we improve transfer volume?

Problem: we transfer data **by value** instead of **by reference.**

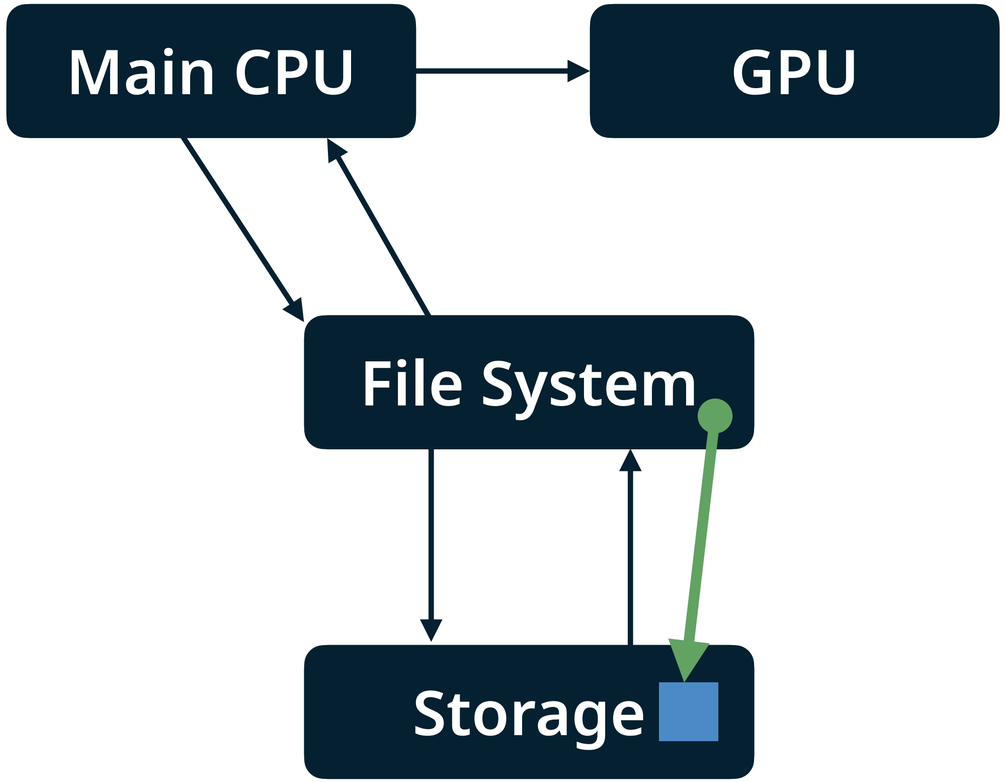
Solution from programming languages: **Pointers**



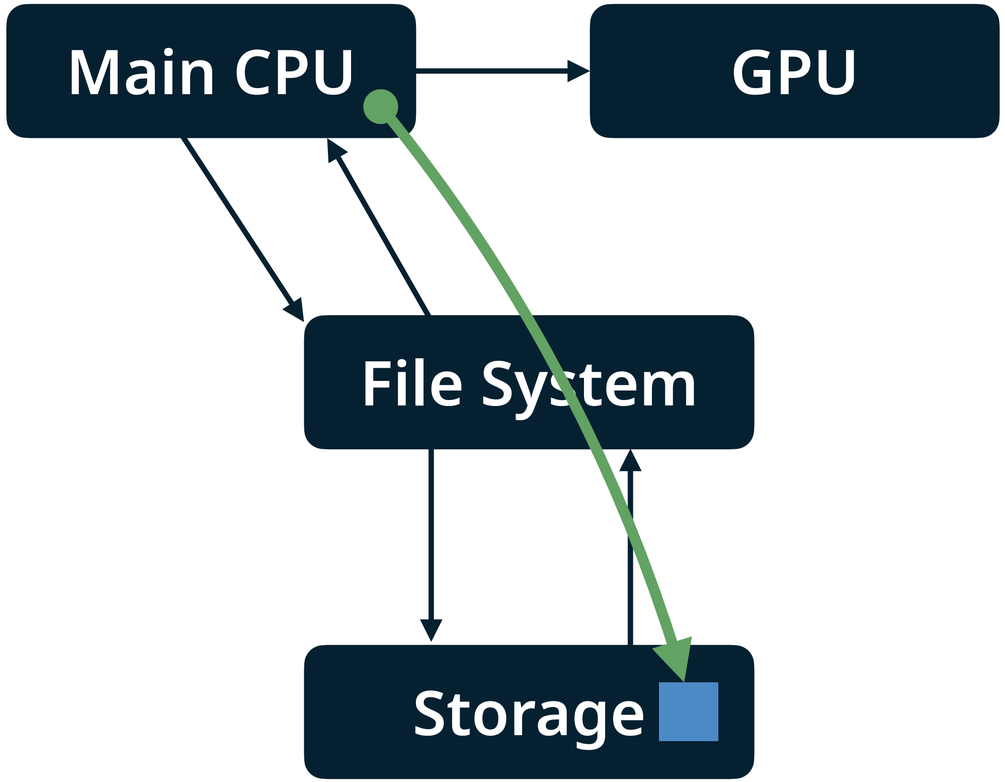
Data Transfer Using Pointers



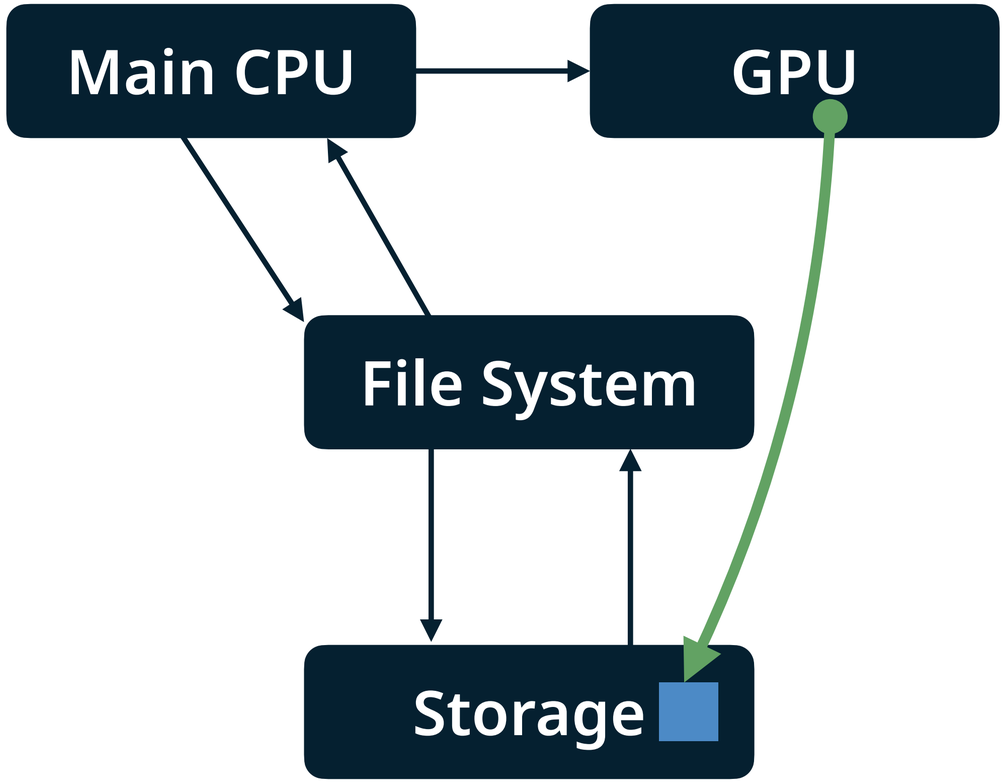
Data Transfer Using Pointers



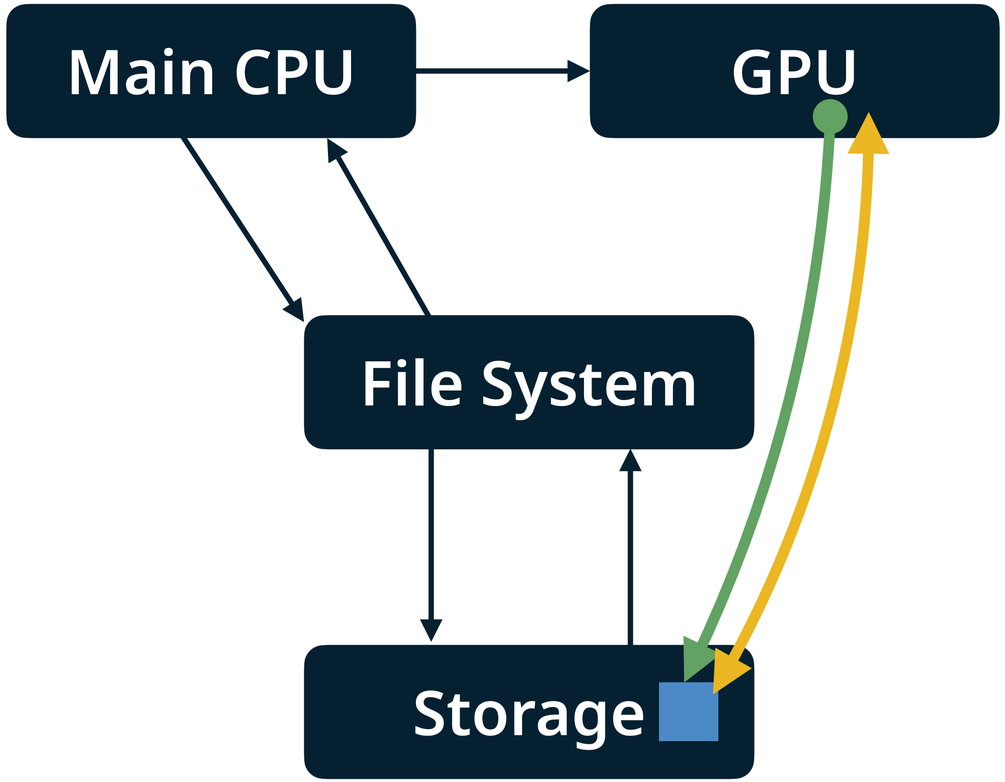
Data Transfer Using Pointers



Data Transfer Using Pointers



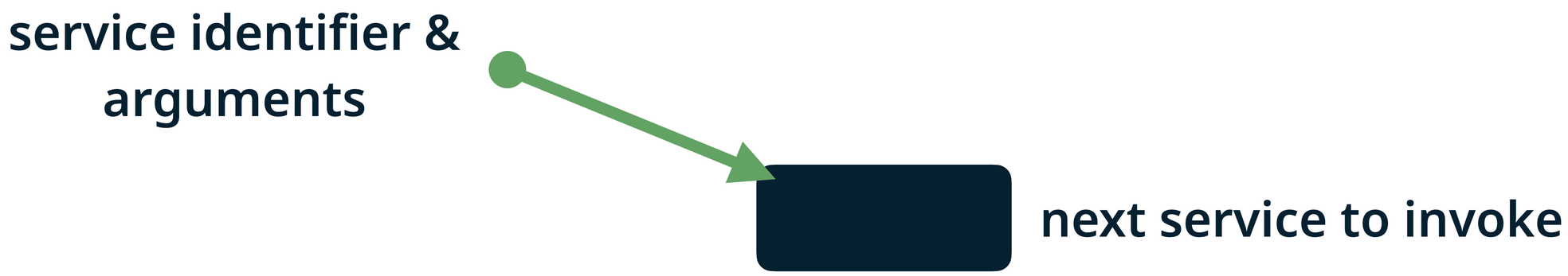
Data Transfer Using Pointers



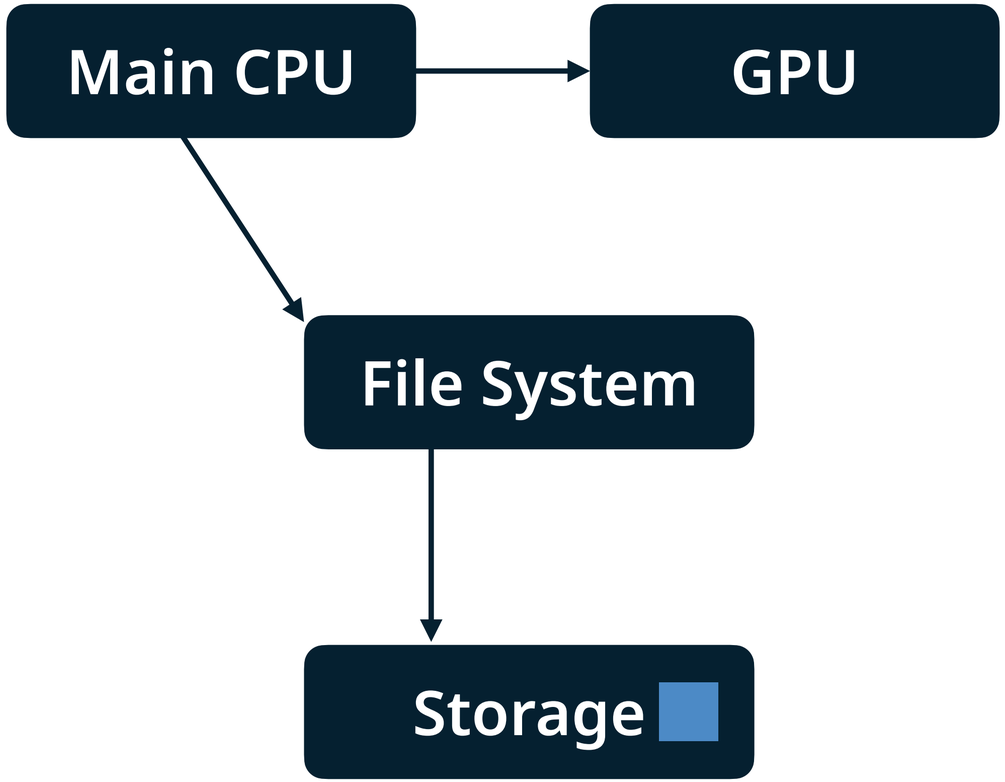
How can we improve transfer latency?

Problem: services return **RPC-style** to the main CPU

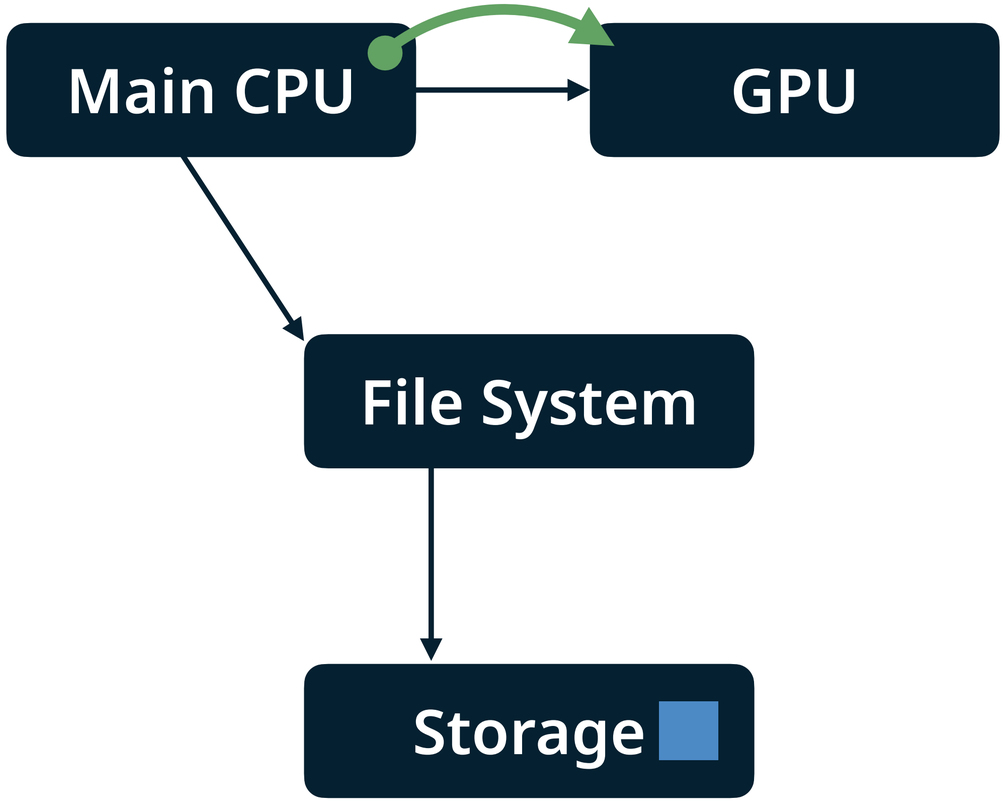
Solution from programming languages: **Continuations**



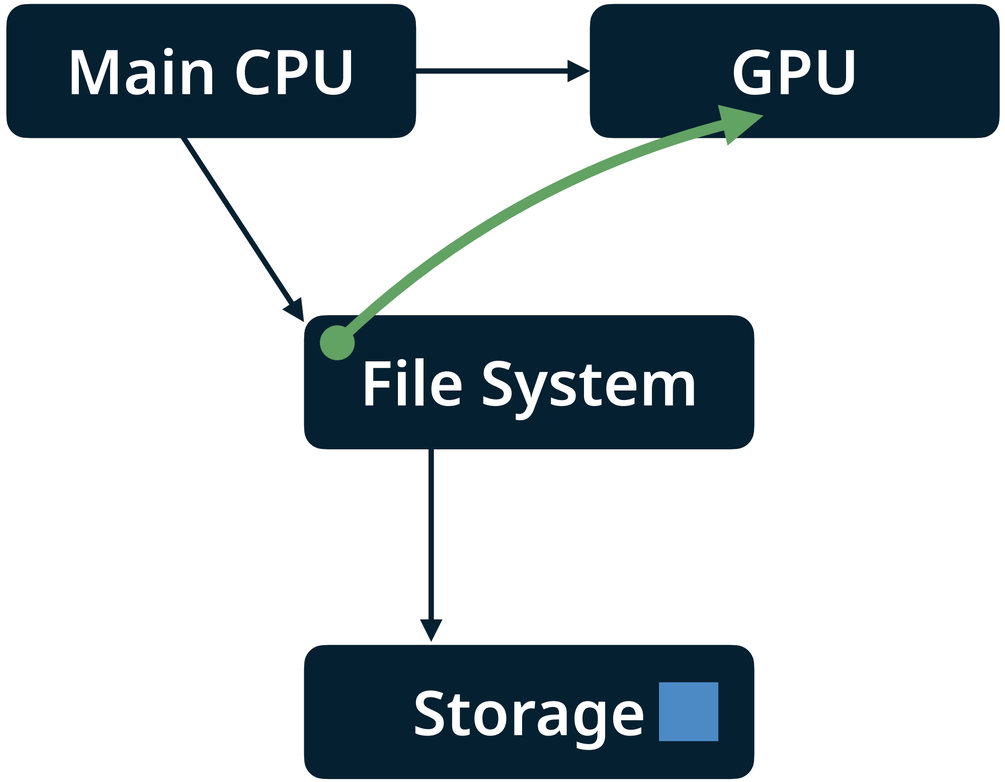
Data Transfer Using Continuations



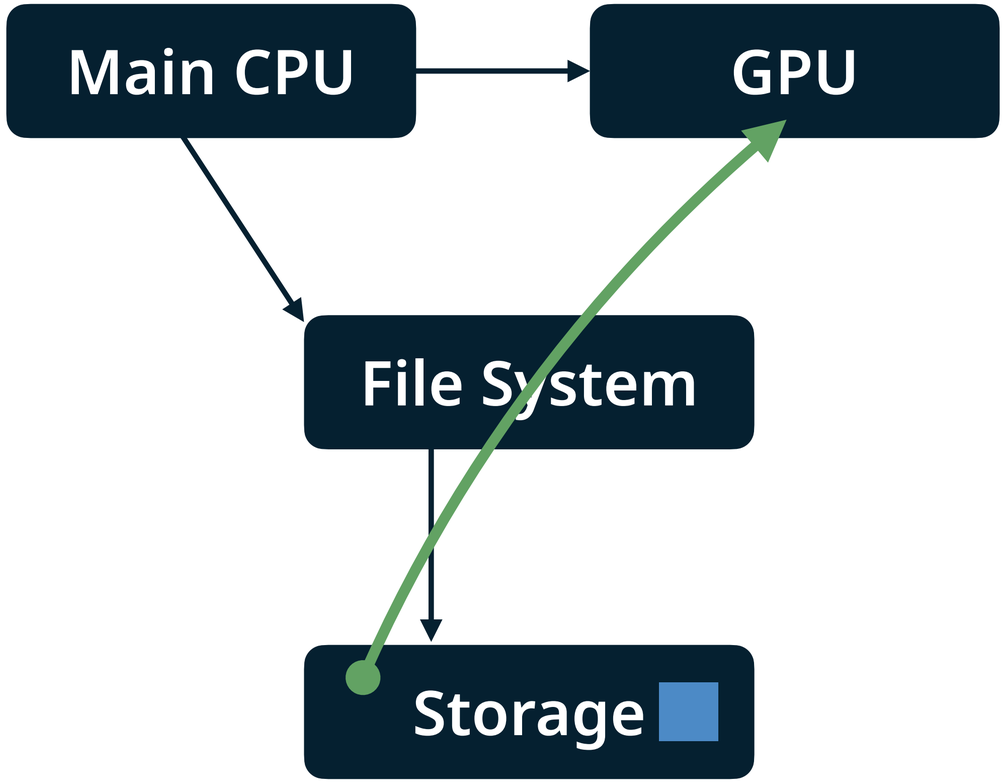
Data Transfer Using Continuations



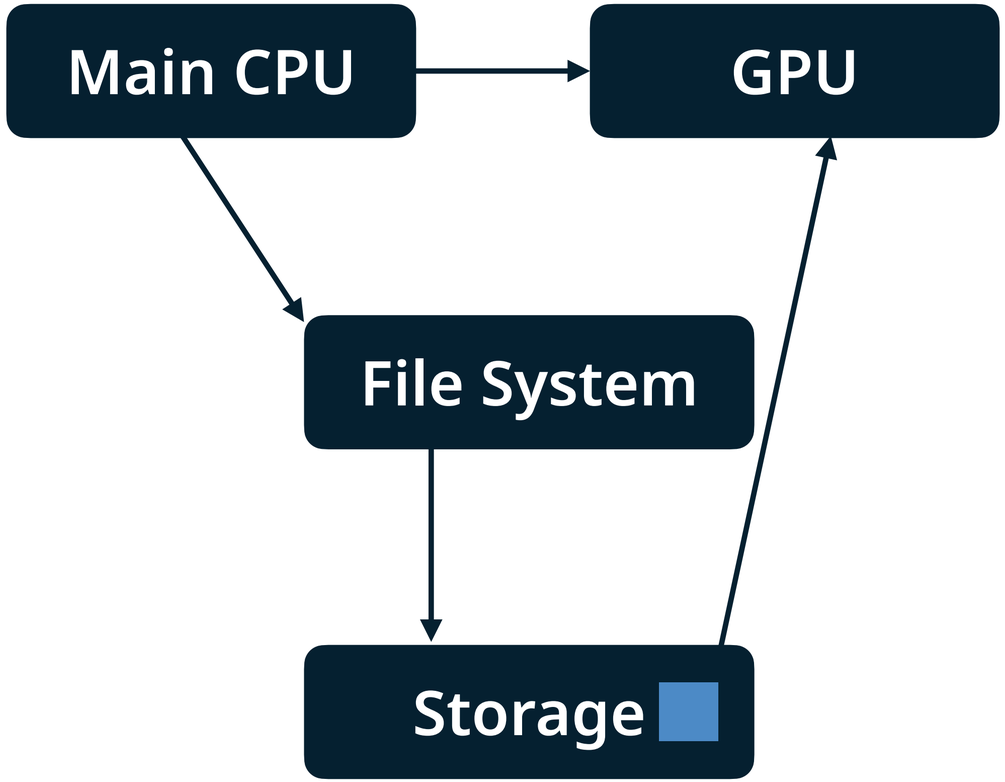
Data Transfer Using Continuations



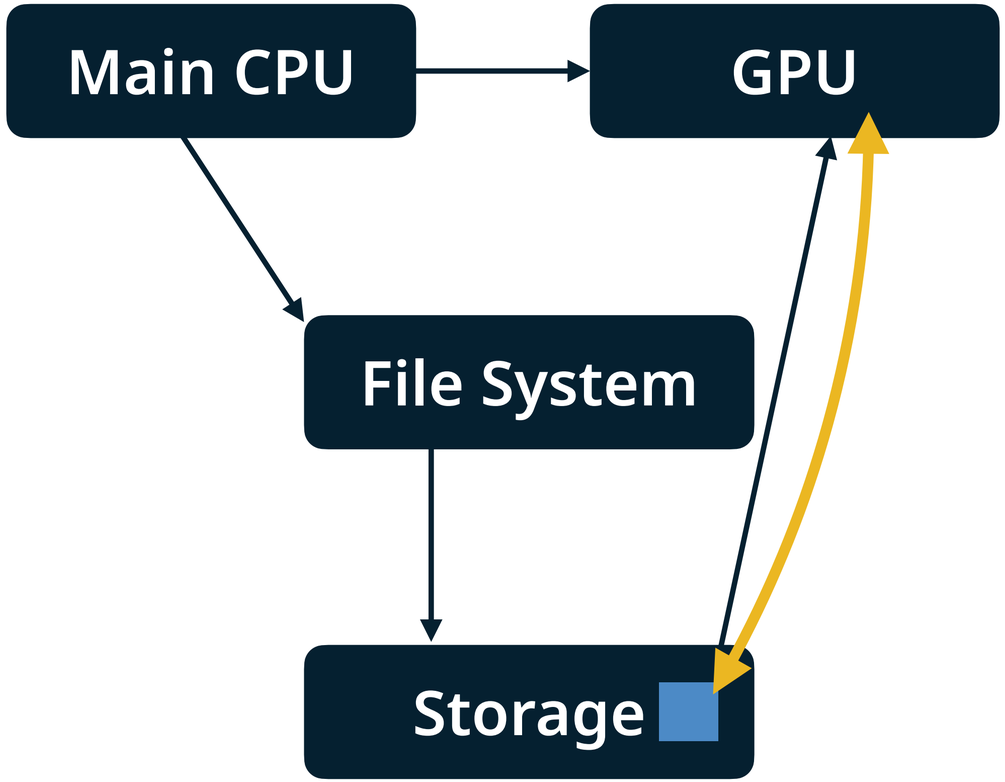
Data Transfer Using Continuations



Data Transfer Using Continuations



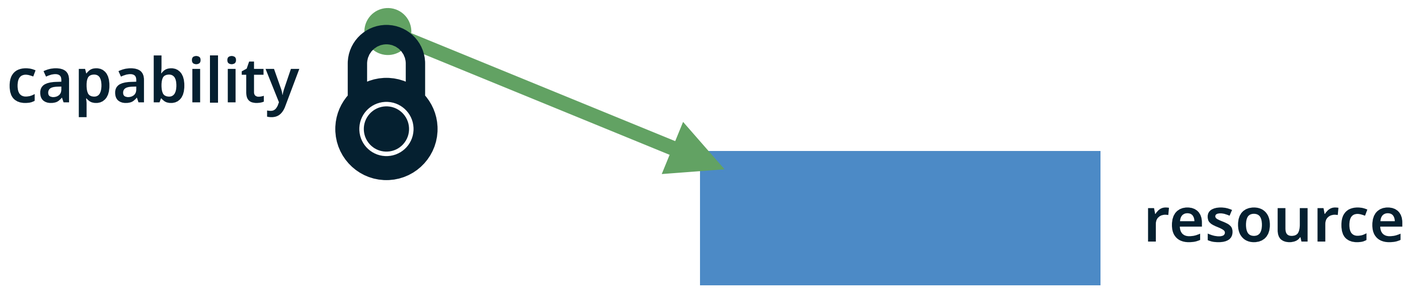
Data Transfer Using Continuations



We created a huge problem!

Every node can **invent or guess** identifiers (pointers).

We need to ensure **validity** of those identifiers!



Capabilities

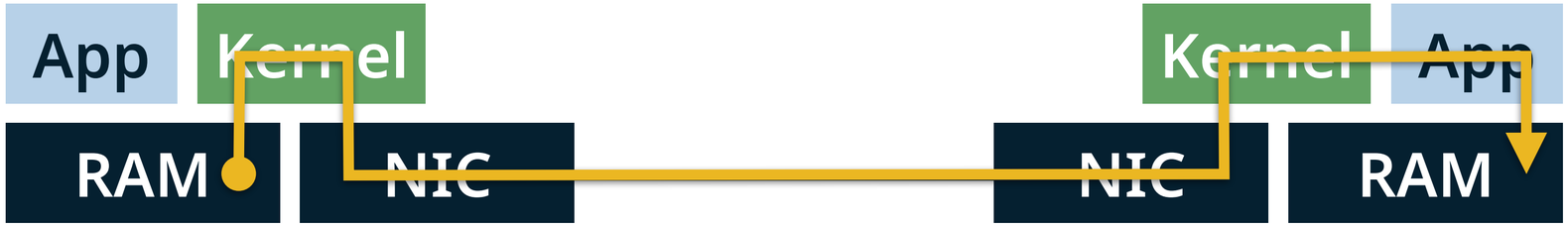
communicable, unforgeable token of authority

Some part of **trusted** infrastructure needs to check access validity.

For datacenters, the solution **must scale.**

The Problem

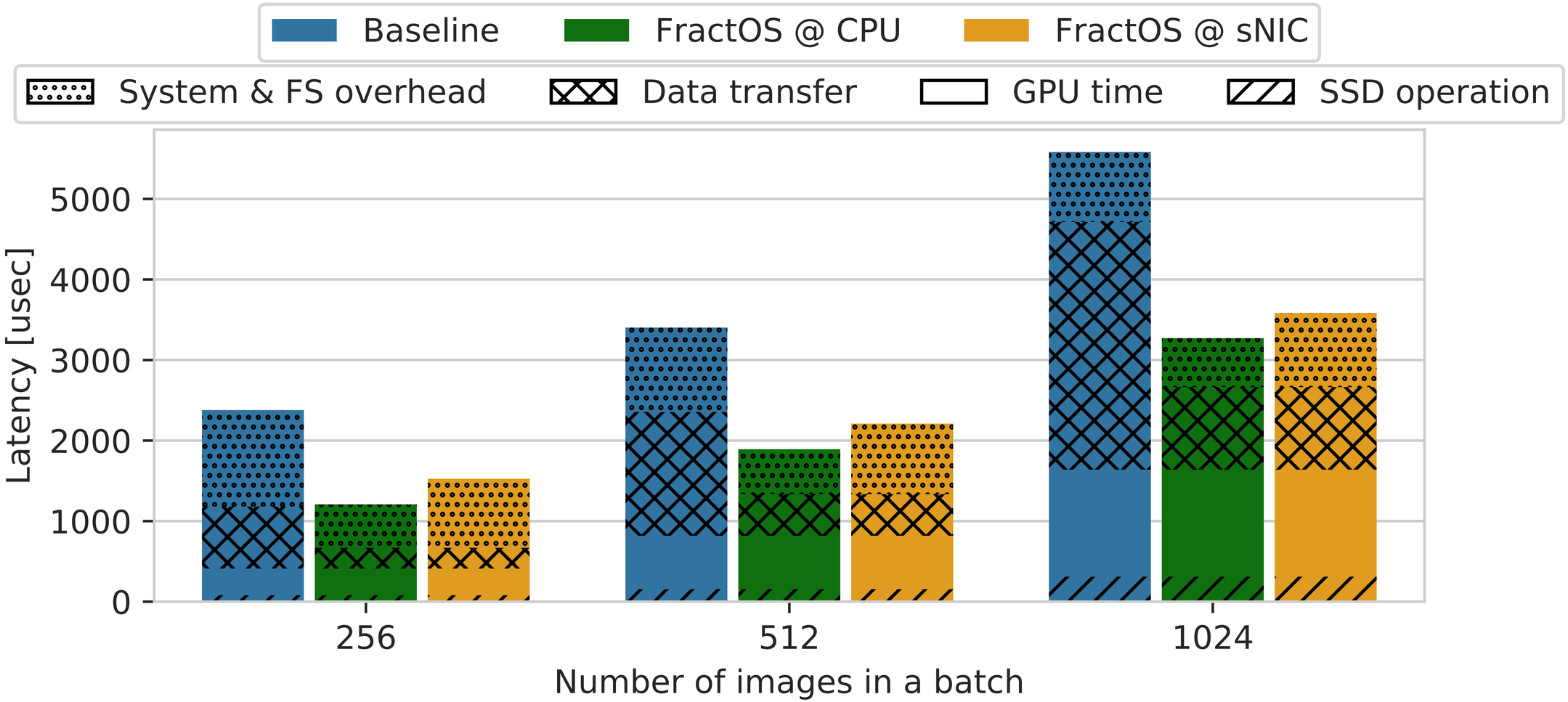
Traditional



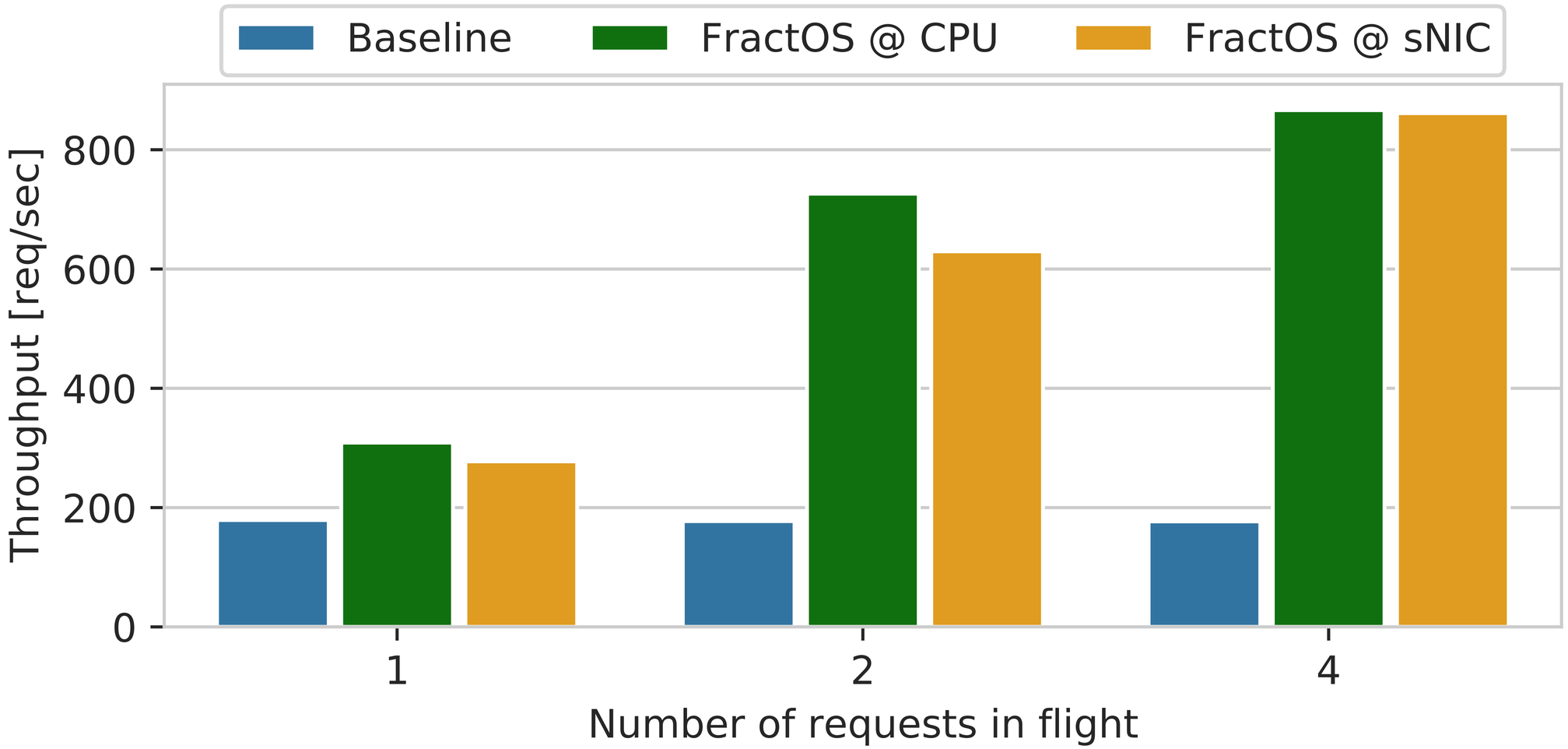
FractOS

M^3

Results: Latency



Results: Throughput



Conclusion

- FractOS brings primitives for a **distributed application control plane**
  - distributed and scalable capability system
  - continuation-based service invocation
- an isolated OS layer enforces those primitives
- significant benefits for latency and throughput of distributed applications
- FractOS allows to **reap the benefits from data center disaggregation**