IMPERIAL

LSDS
Large-Scale Data & Systems Group

Enabling Cloud Scale Distributed Capabilities

Otto White[1], Yaoxin Jing[1], Adrien Ghosn[2], Anjo Vahldiek-Oberwagner[3], Mona Vij[3], Michael Steiner[3], Lluis Vilanova[1]

HCDS '25

<otto.white20@imperial.ac.uk>

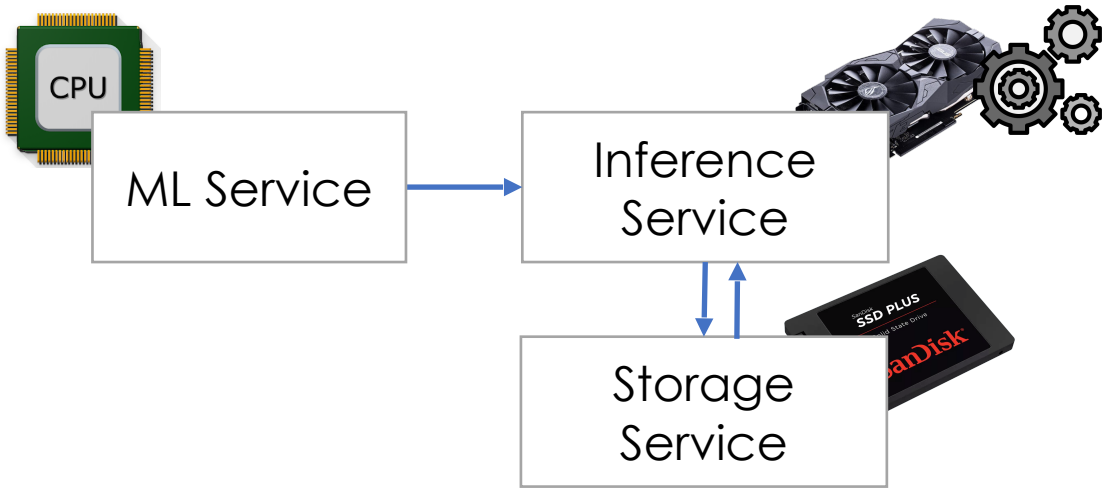
IMPERIAL[1]

[2]

intel[3] labs

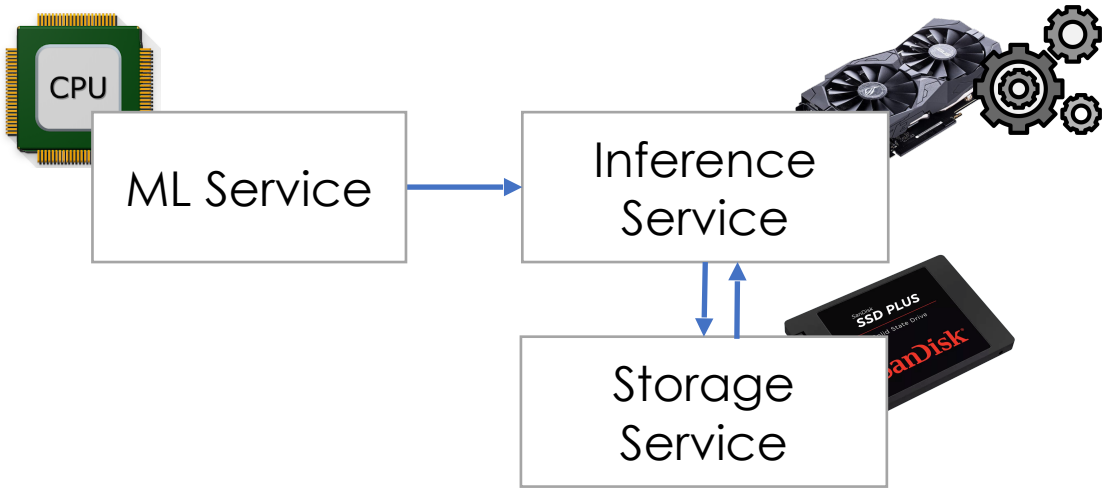
Security in Disaggregated Cloud

Security in Disaggregated Cloud



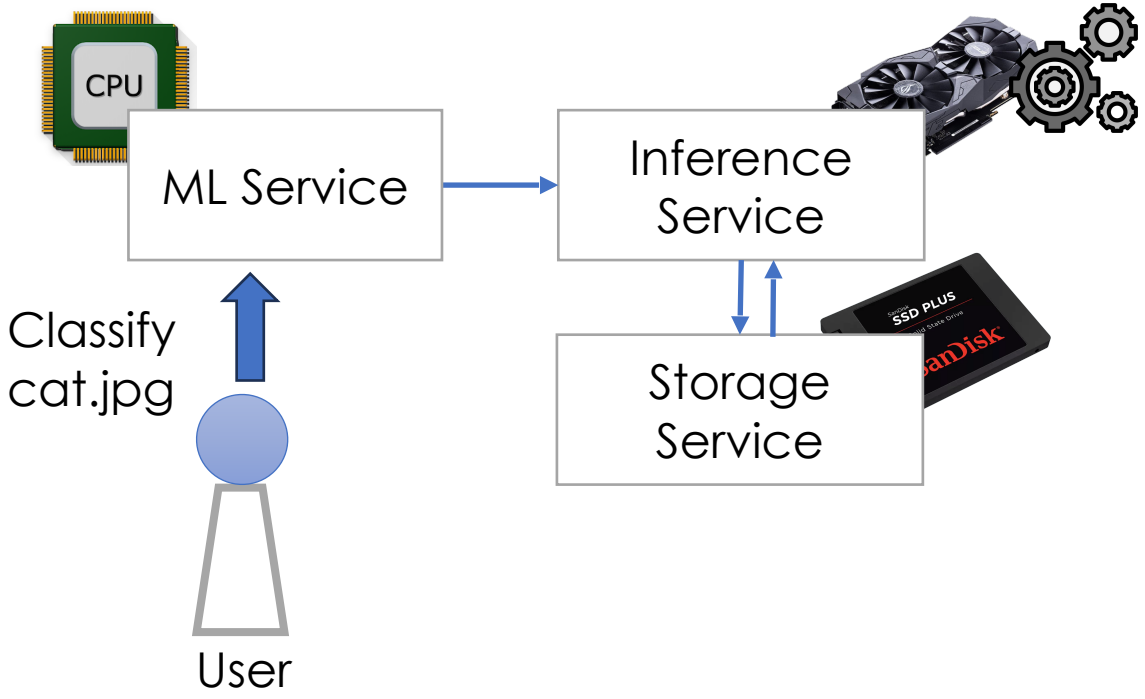
- Disaggregated Service-Oriented Architecture

Security in Disaggregated Cloud



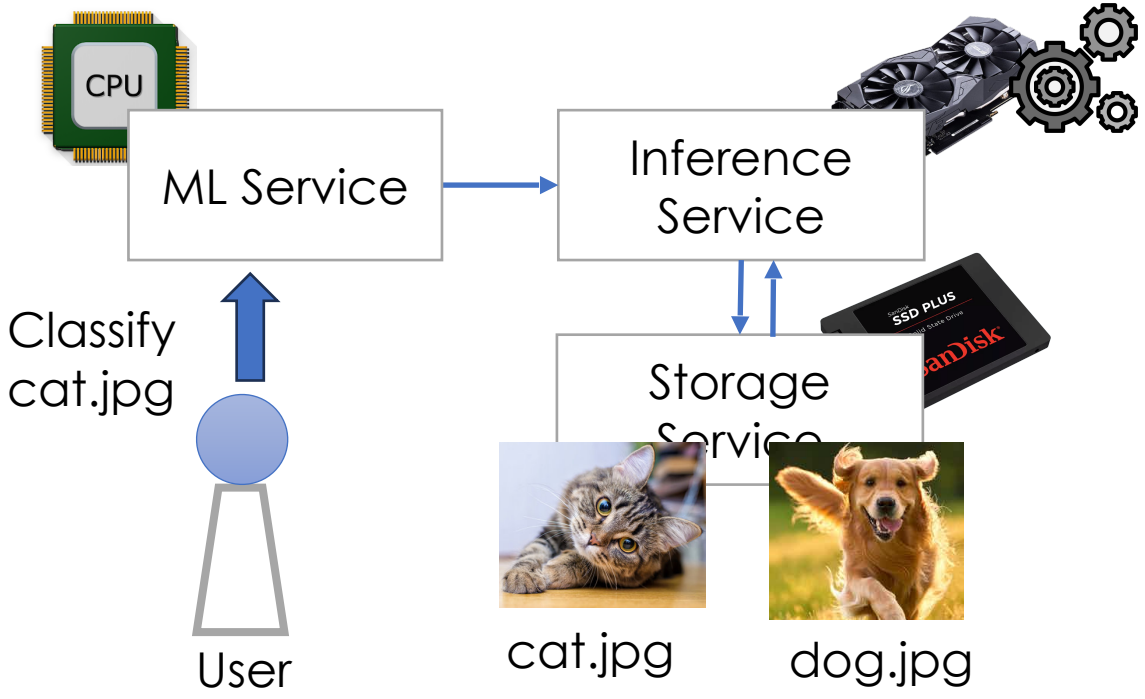
- Disaggregated Service-Oriented Architecture
- Minimizing privilege is essential

Security in Disaggregated Cloud



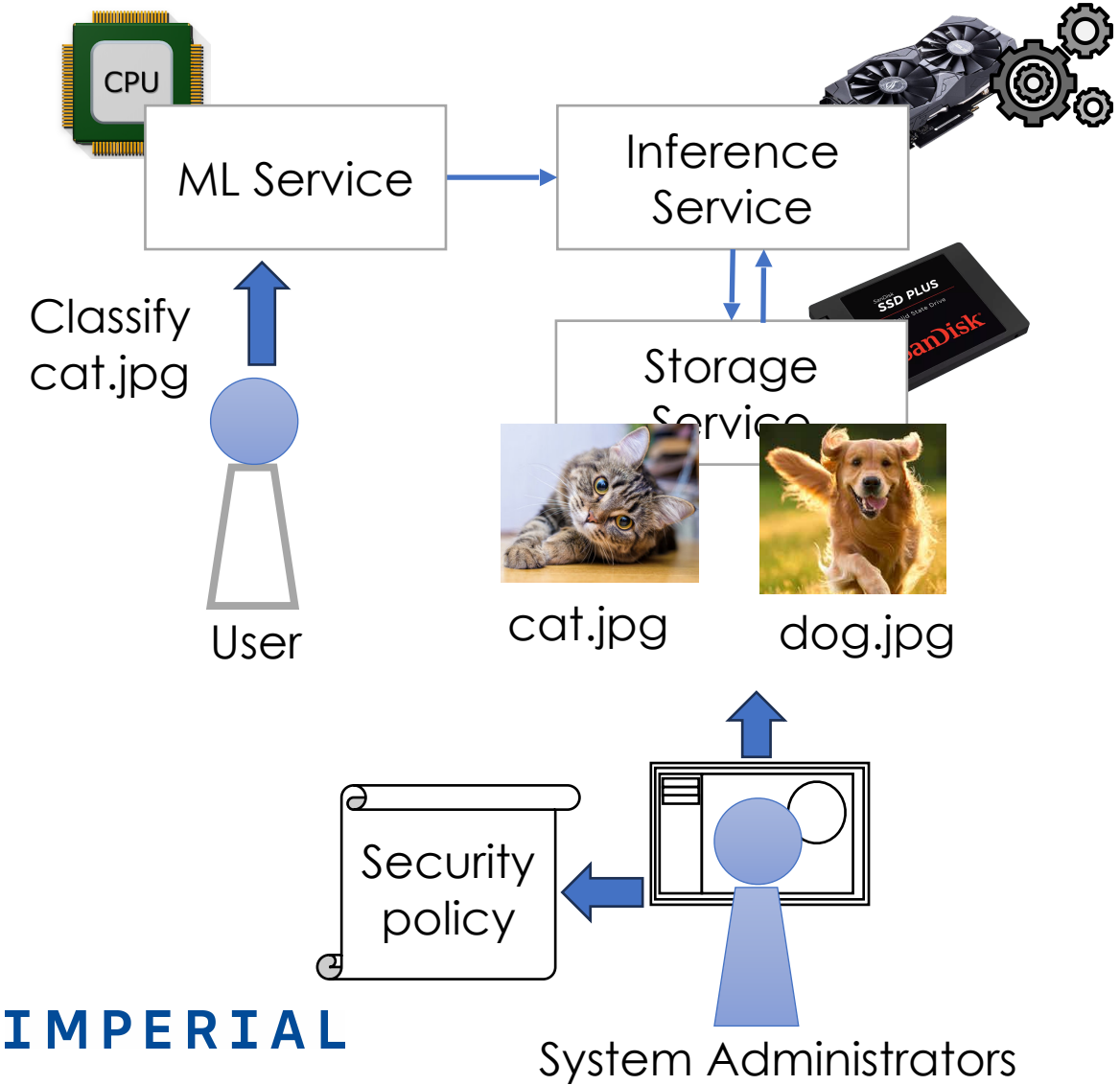
- Disaggregated Service-Oriented Architecture
- Minimizing privilege is essential

Security in Disaggregated Cloud



- Disaggregated Service-Oriented Architecture
- Minimizing privilege is essential

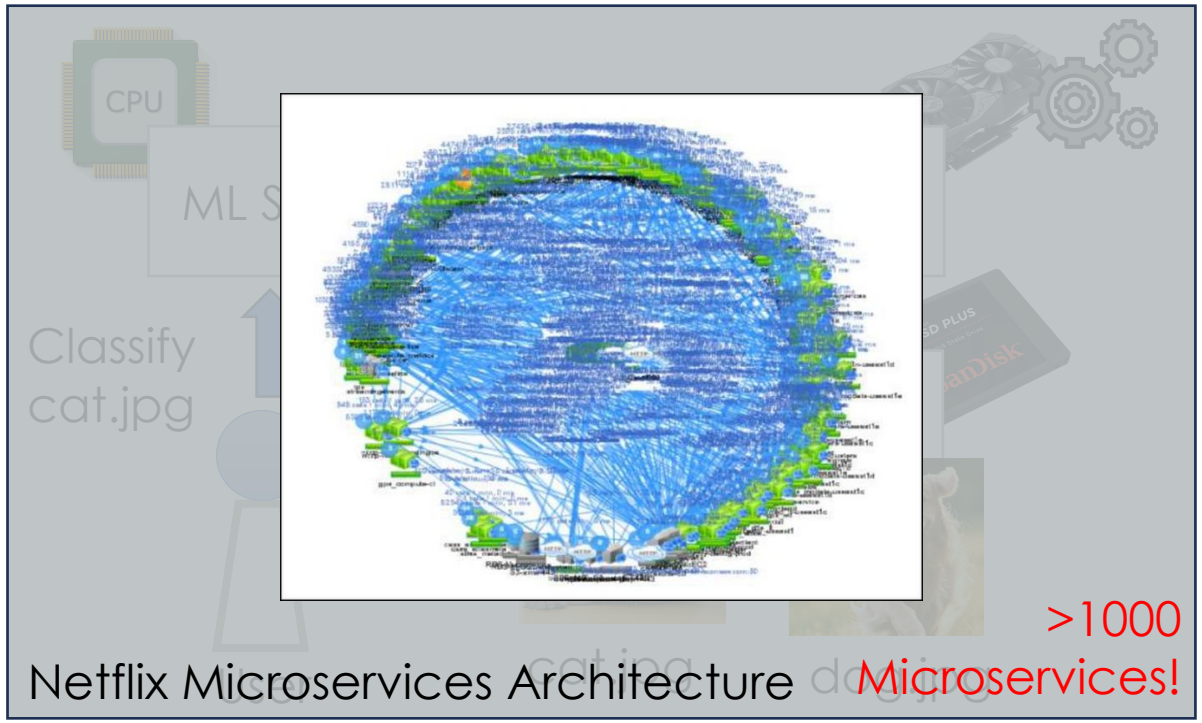
Security in Disaggregated Cloud



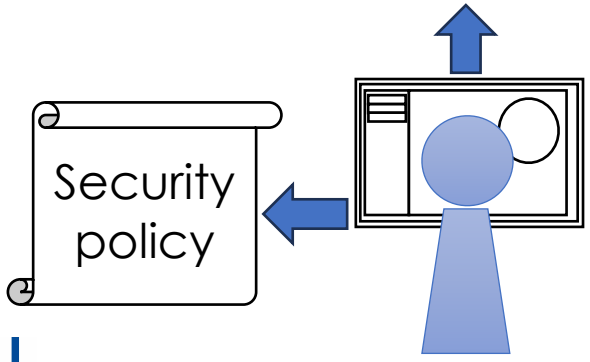
- Disaggregated Service-Oriented Architecture
- Minimizing privilege is essential

IMPERIAL

Security in Disaggregated Cloud

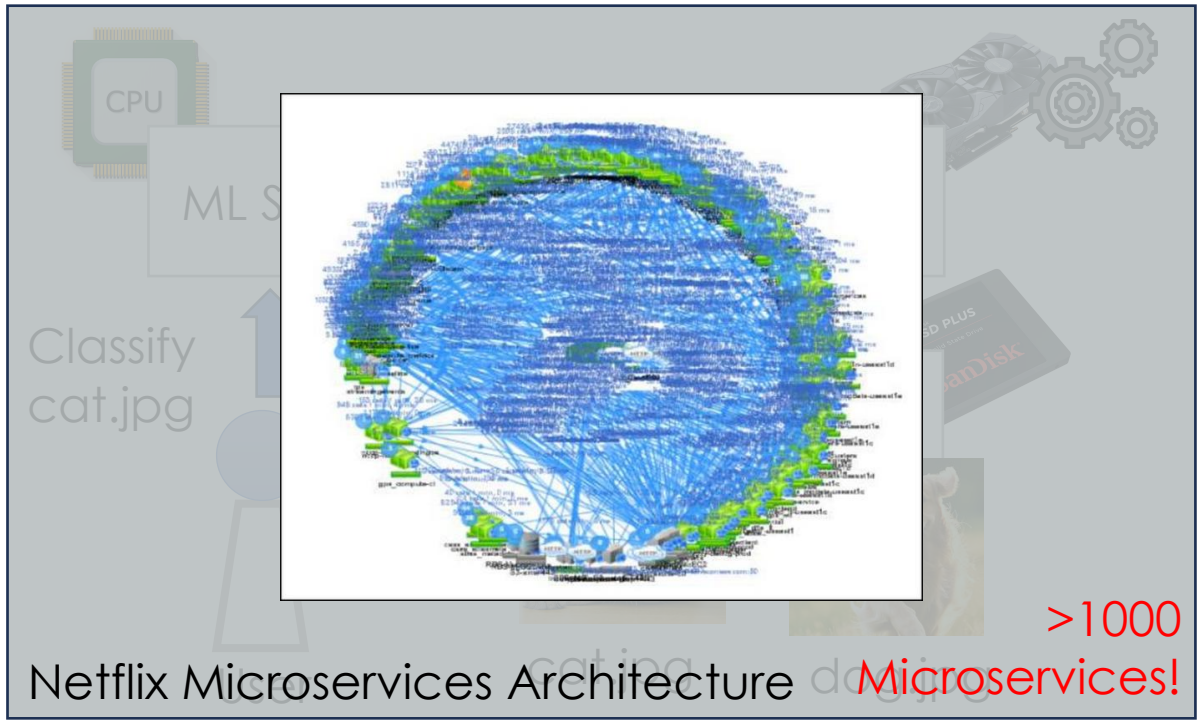


- Disaggregated Service-Oriented Architecture
- Minimizing privilege is essential



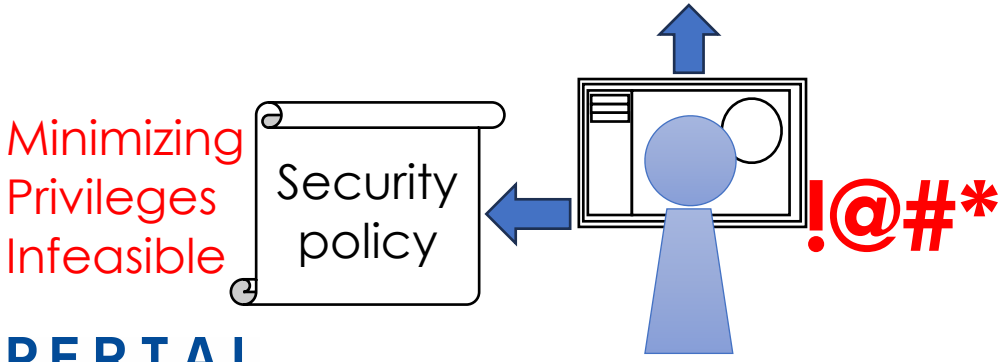
IMPERIAL

Security in Disaggregated Cloud



Netflix Microservices Architecture

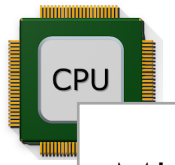
- Disaggregated Service-Oriented Architecture
- Minimizing privilege is essential
- Policies don't scale to complex systems -> over-privilege



System Administrators

IMPERIAL

Capabilities for Disaggregated Cloud

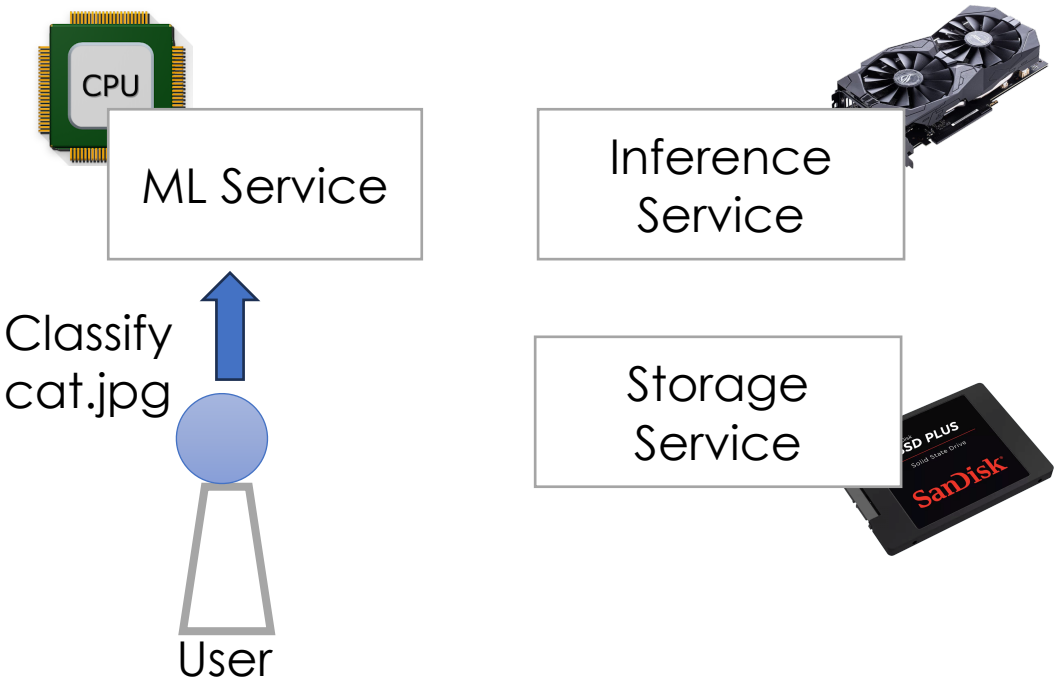


ML Service

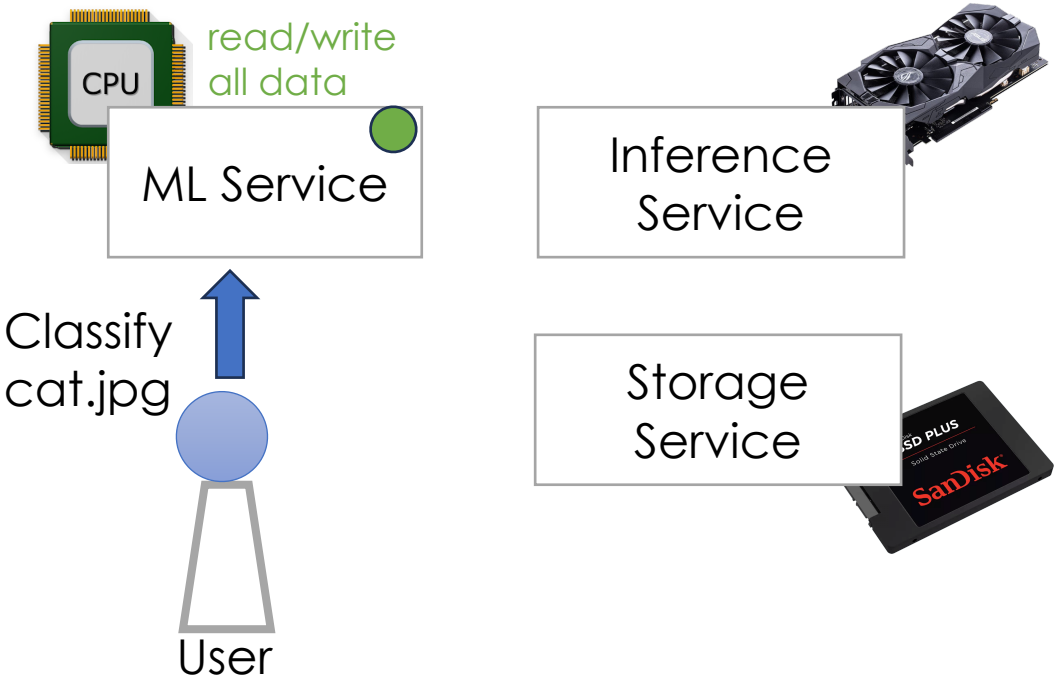
Inference Service

Storage Service

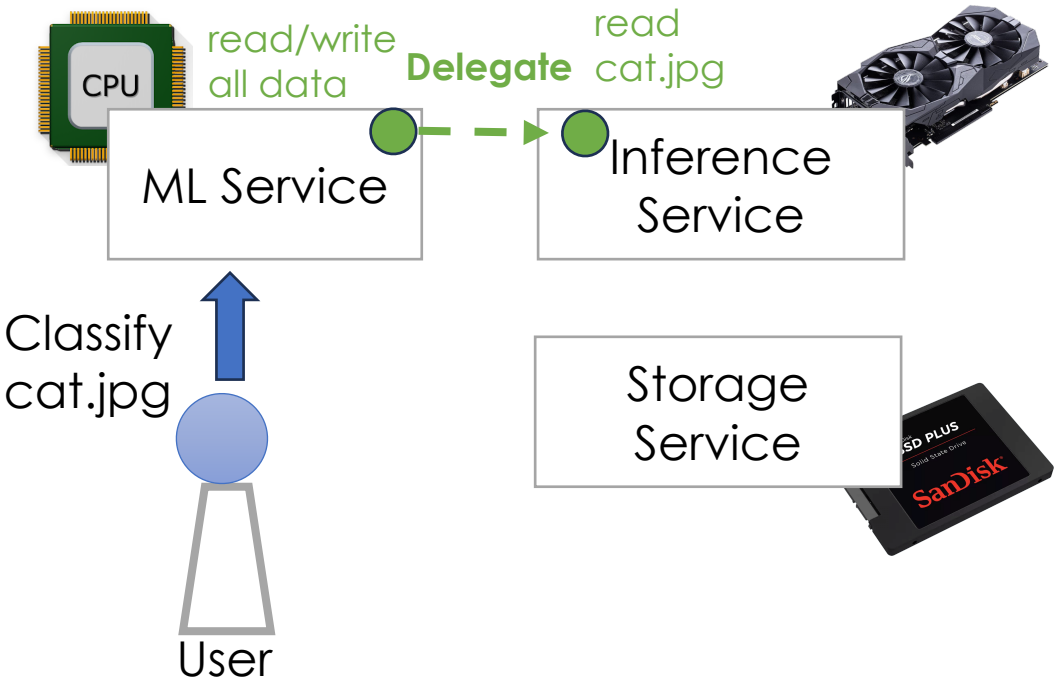
Capabilities for Disaggregated Cloud



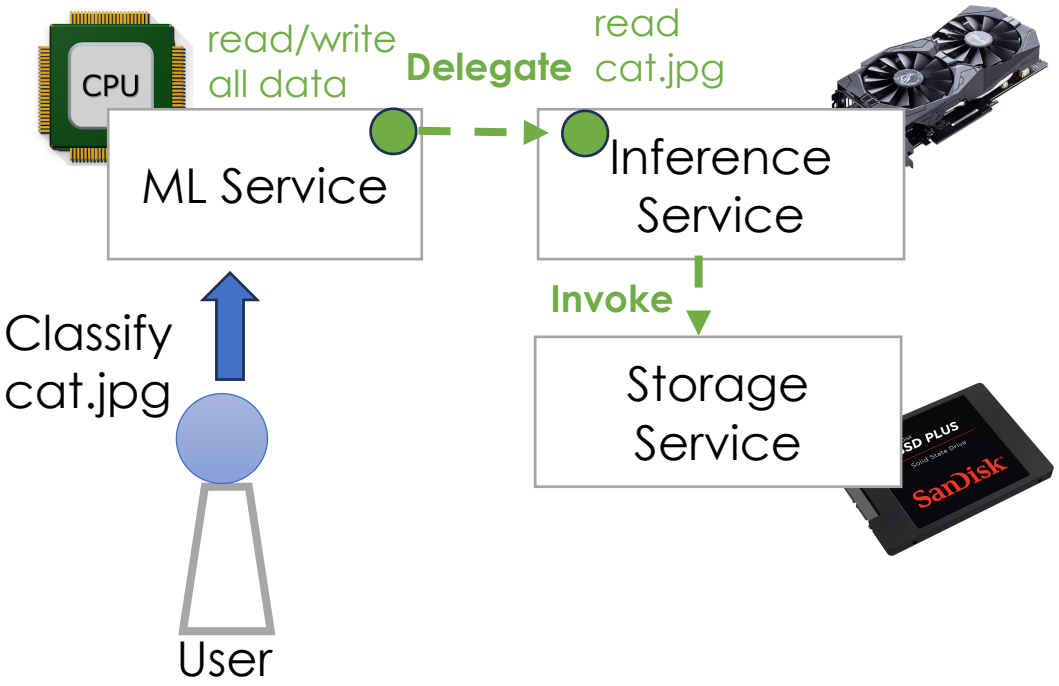
Capabilities for Disaggregated Cloud



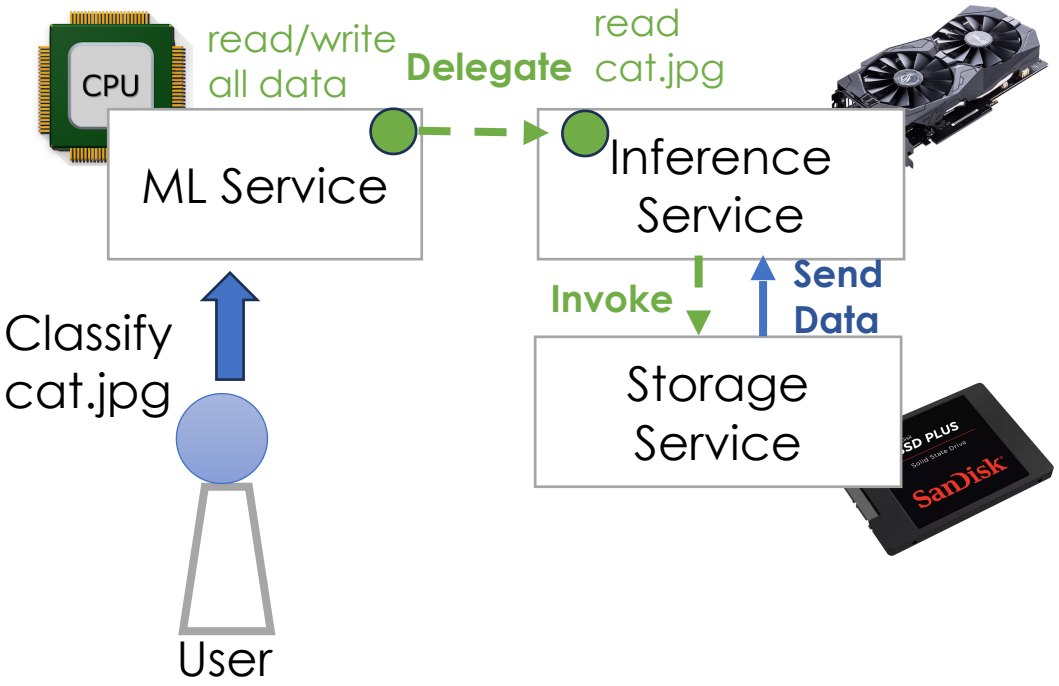
Capabilities for Disaggregated Cloud



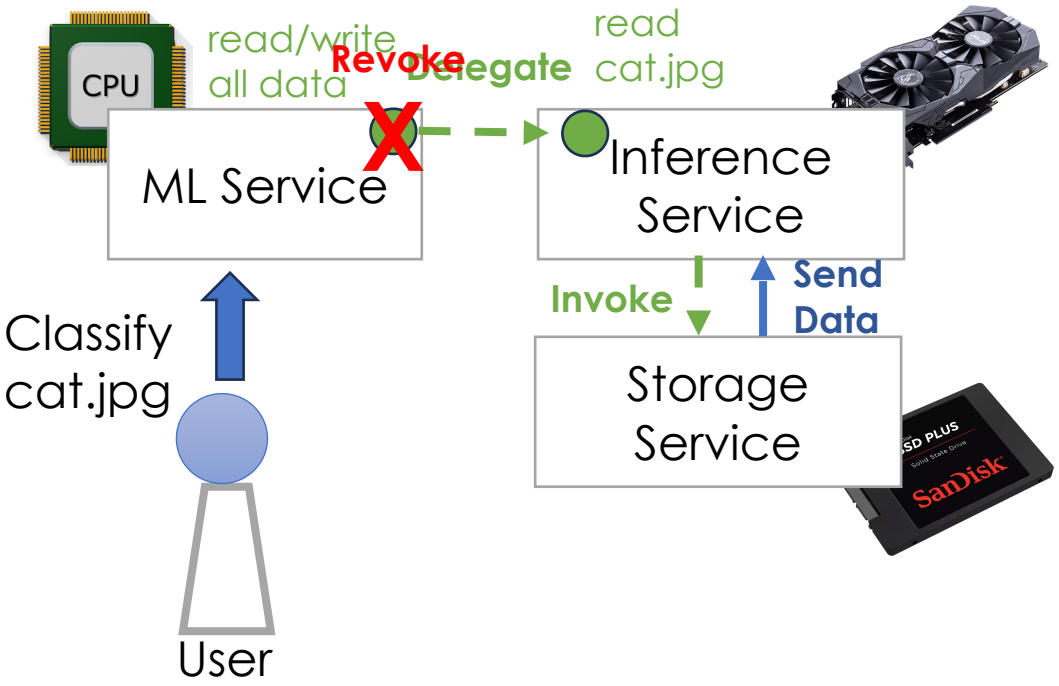
Capabilities for Disaggregated Cloud



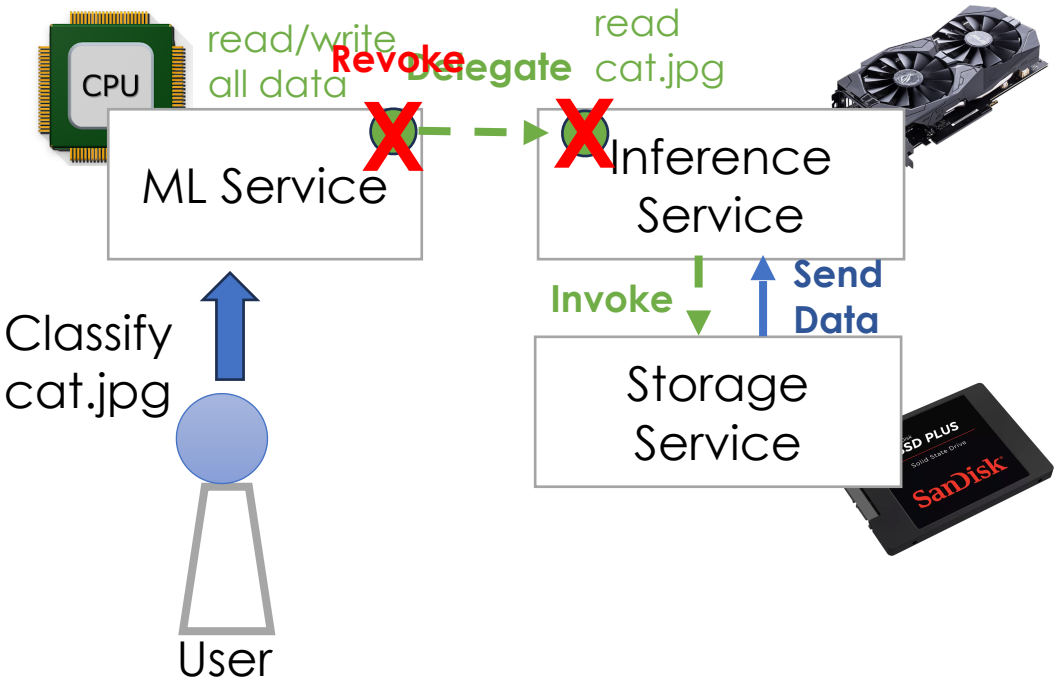
Capabilities for Disaggregated Cloud



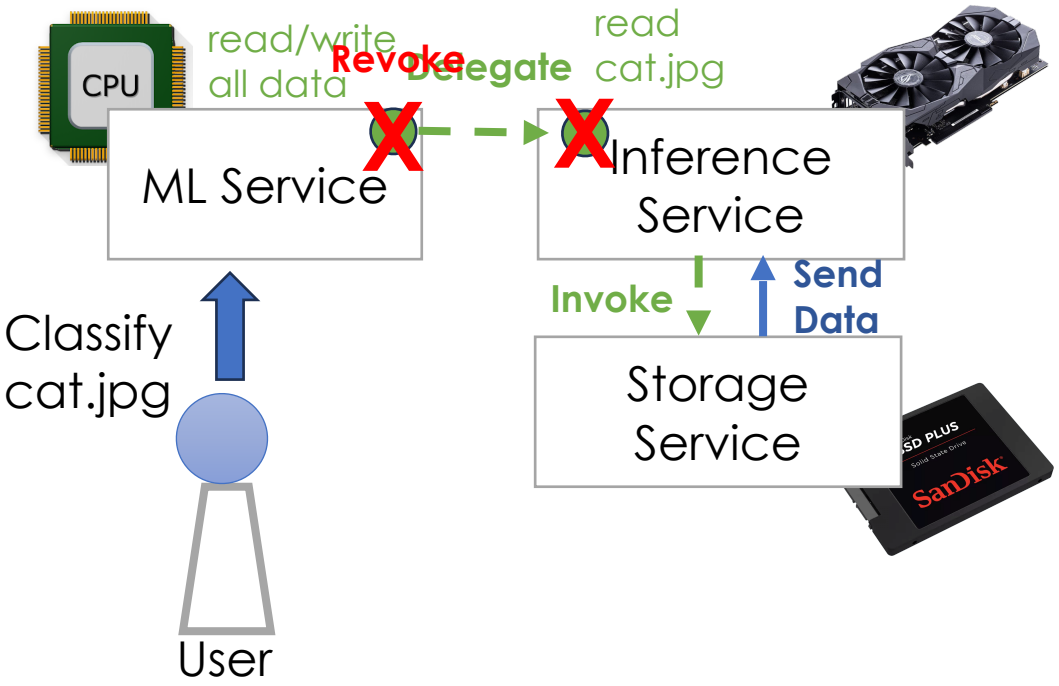
Capabilities for Disaggregated Cloud



Capabilities for Disaggregated Cloud

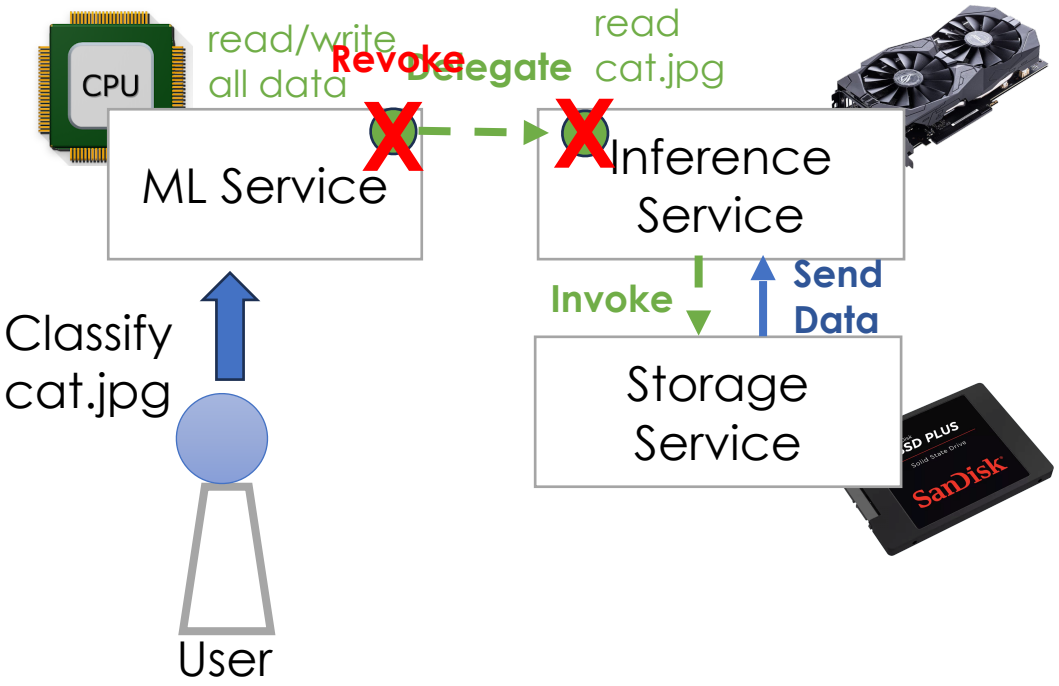


Capabilities for Disaggregated Cloud



- Application controlled permissions

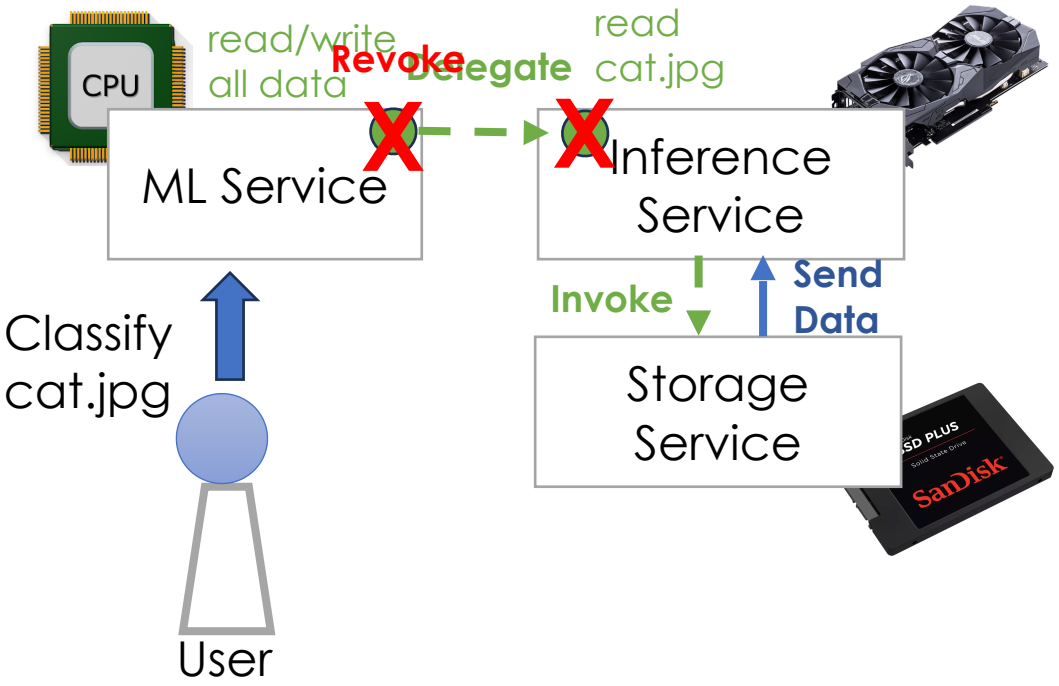
Capabilities for Disaggregated Cloud



- Application controlled permissions

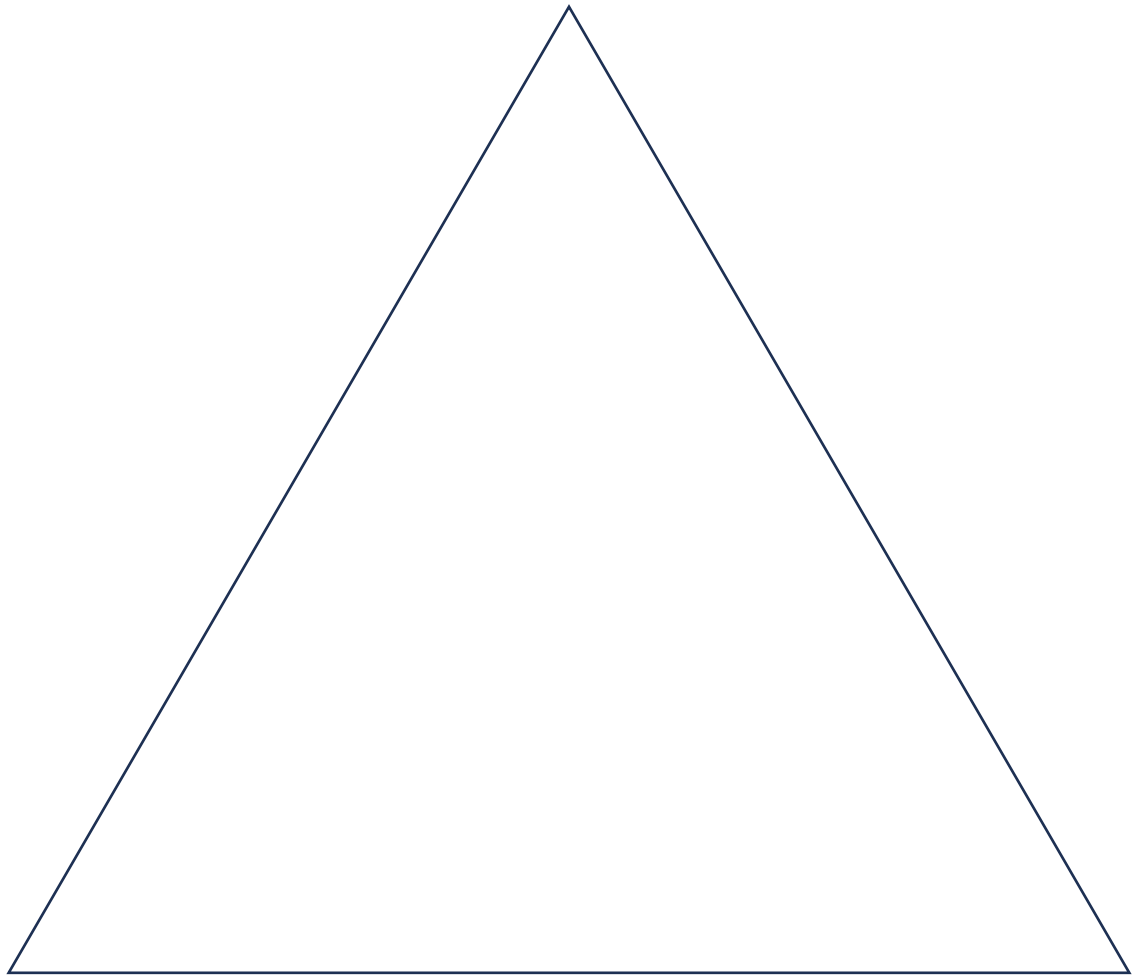
  – Policy is distributed

Capabilities for Disaggregated Cloud

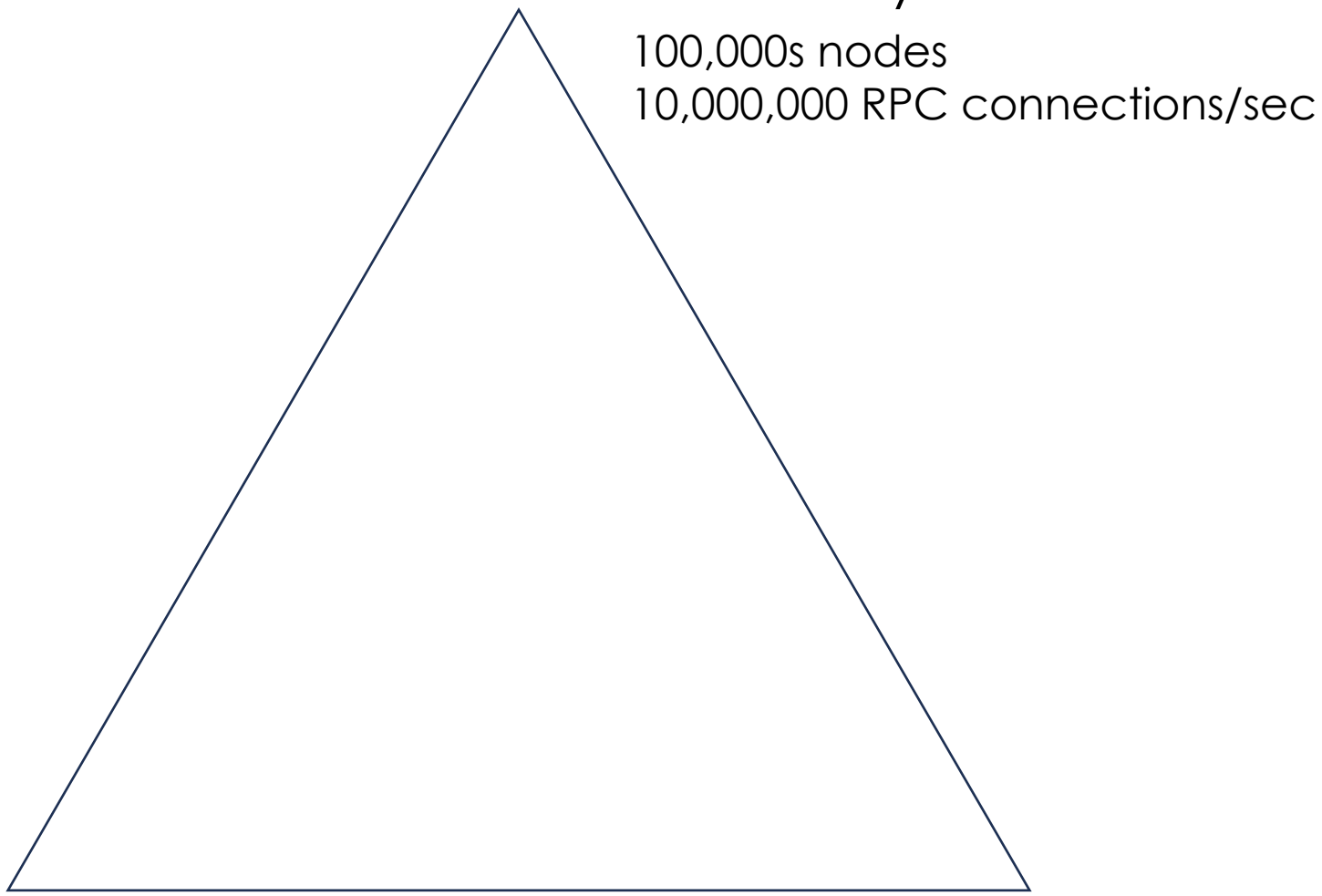


- Application controlled permissions
  - Policy is distributed
  - Privilege minimization scales better to complex systems

No Capability Systems Are Cloud Ready



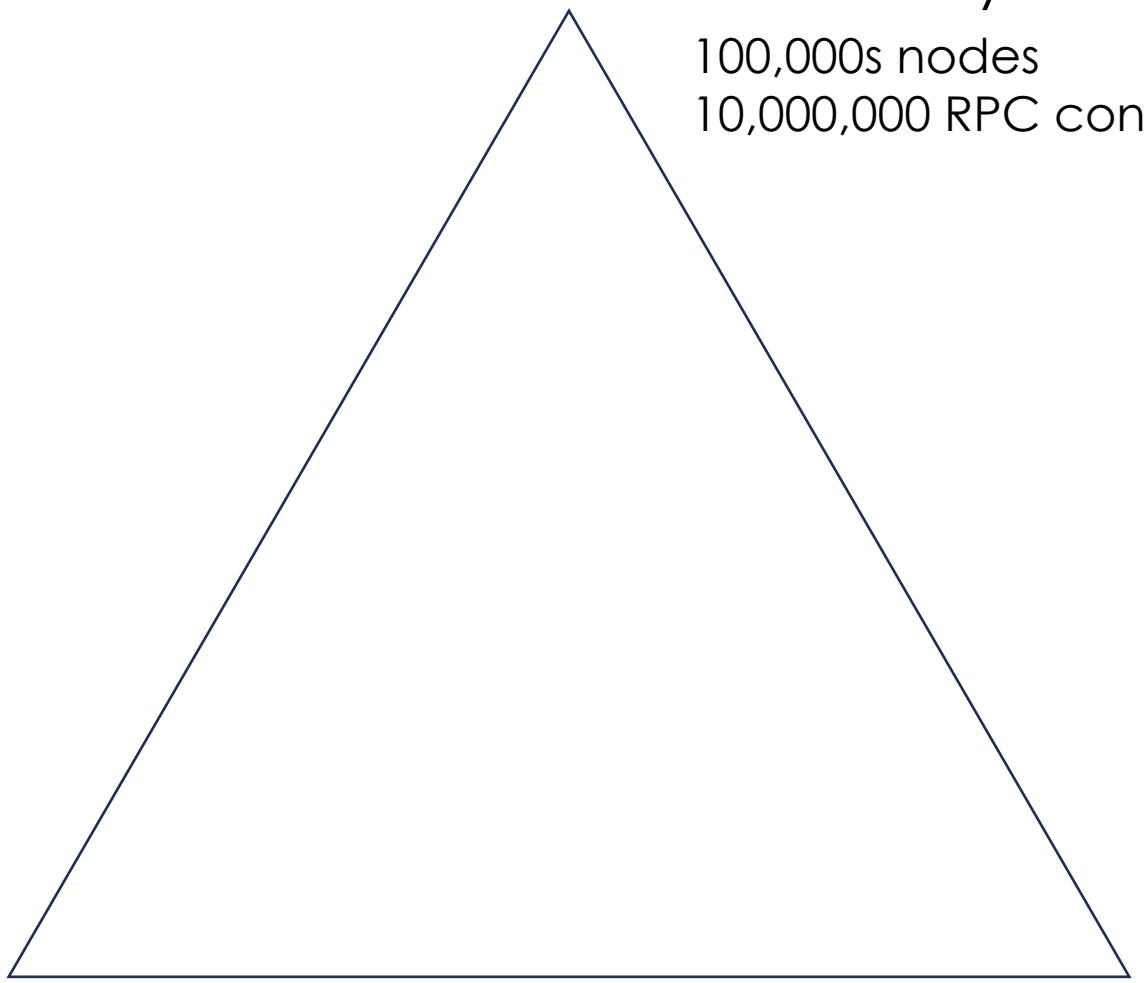
No Capability Systems Are Cloud Ready



No Capability Systems Are Cloud Ready

Scalability

100,000s nodes
10,000,000 RPC connections/sec



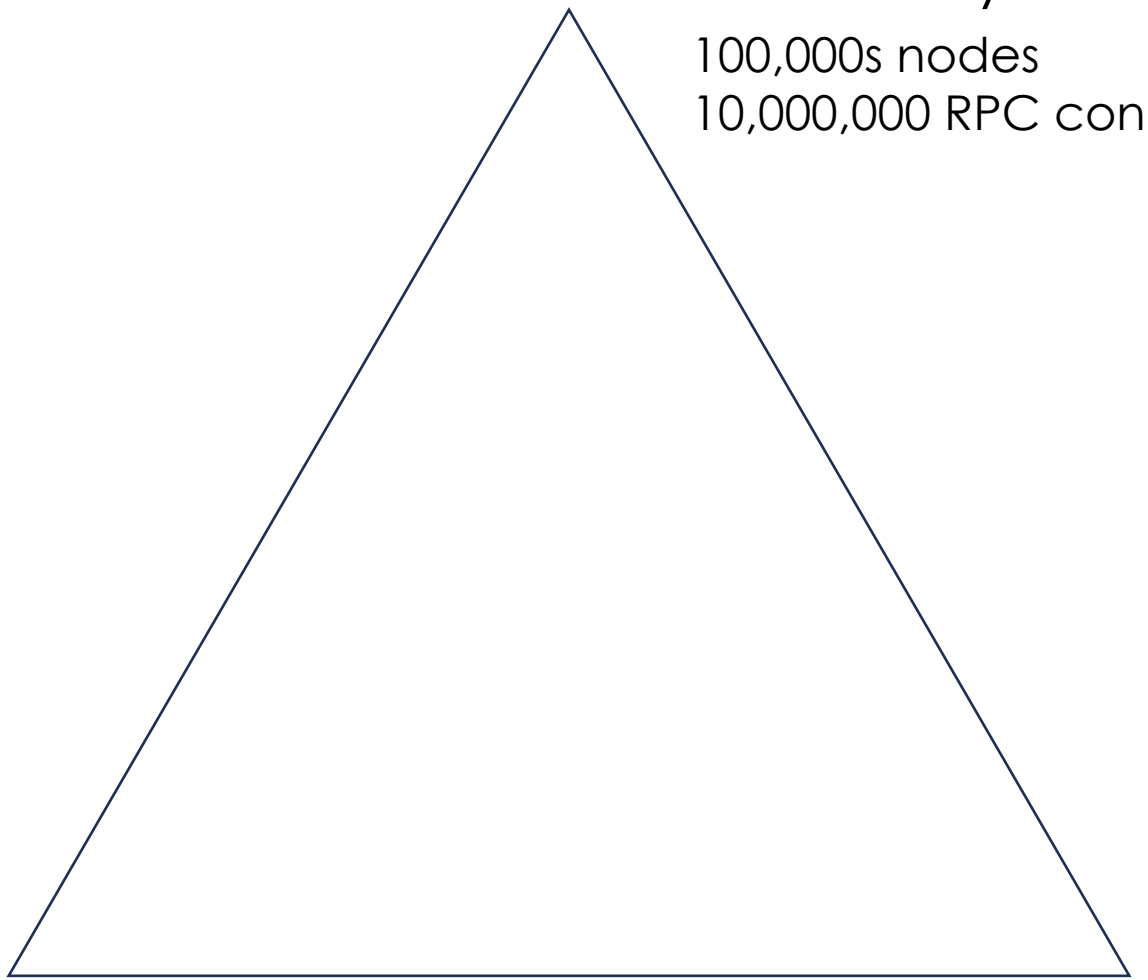
Critical-path Performance

1-2 µs-scale RPCs

No Capability Systems Are Cloud Ready

Scalability

100,000s nodes
10,000,000 RPC connections/sec



Critical-path Performance

1-2 µs-scale RPCs

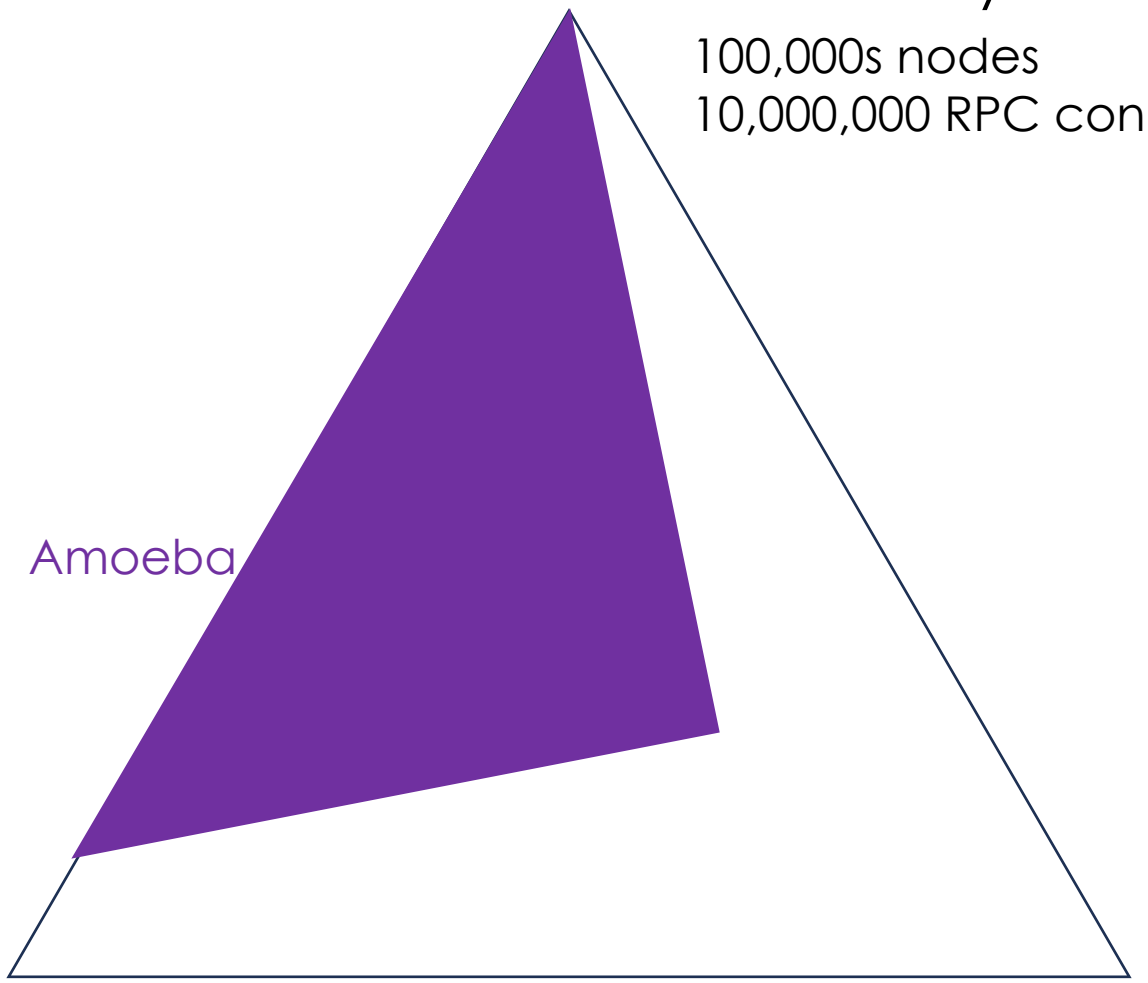
Fault-tolerance

<17.5 failures/node day
~35,000 failures on 2k nodes

No Capability Systems Are Cloud Ready

Scalability

100,000s nodes
10,000,000 RPC connections/sec



Critical-path Performance

1-2 µs-scale RPCs

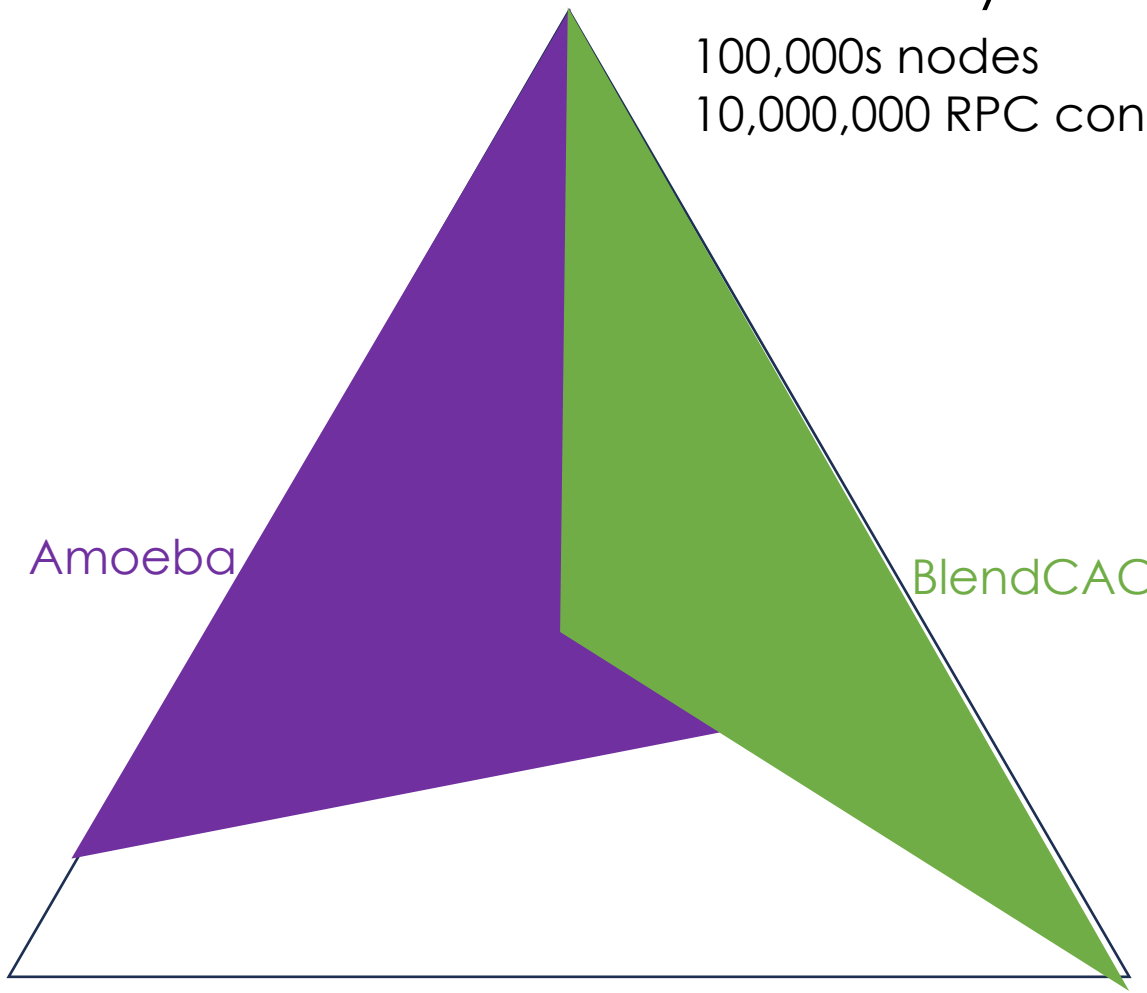
Fault-tolerance

<17.5 failures/node day
~35,000 failures on 2k nodes

No Capability Systems Are Cloud Ready

Scalability

100,000s nodes
10,000,000 RPC connections/sec



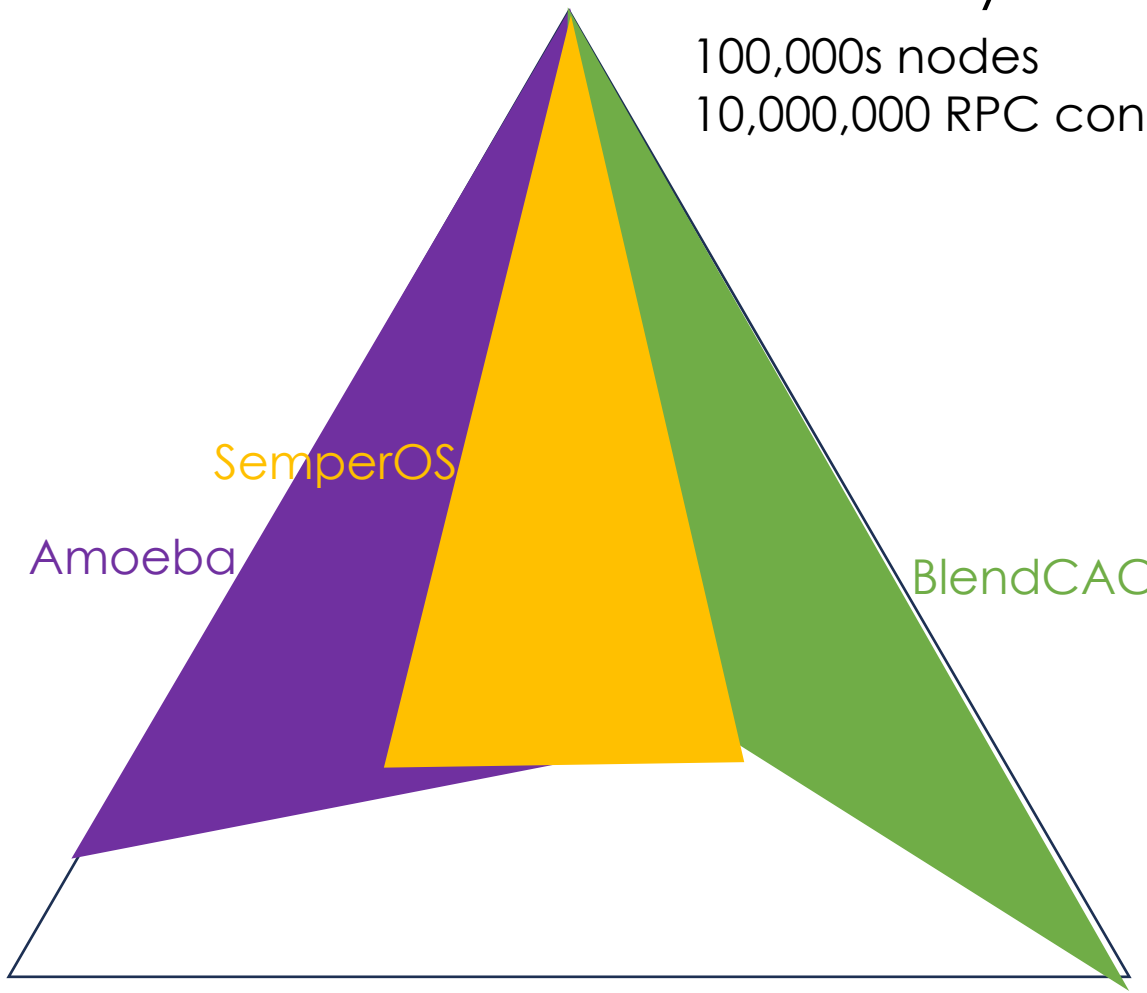
Critical-path Performance

1-2 µs-scale RPCs

Fault-tolerance

<17.5 failures/node day
~35,000 failures on 2k nodes

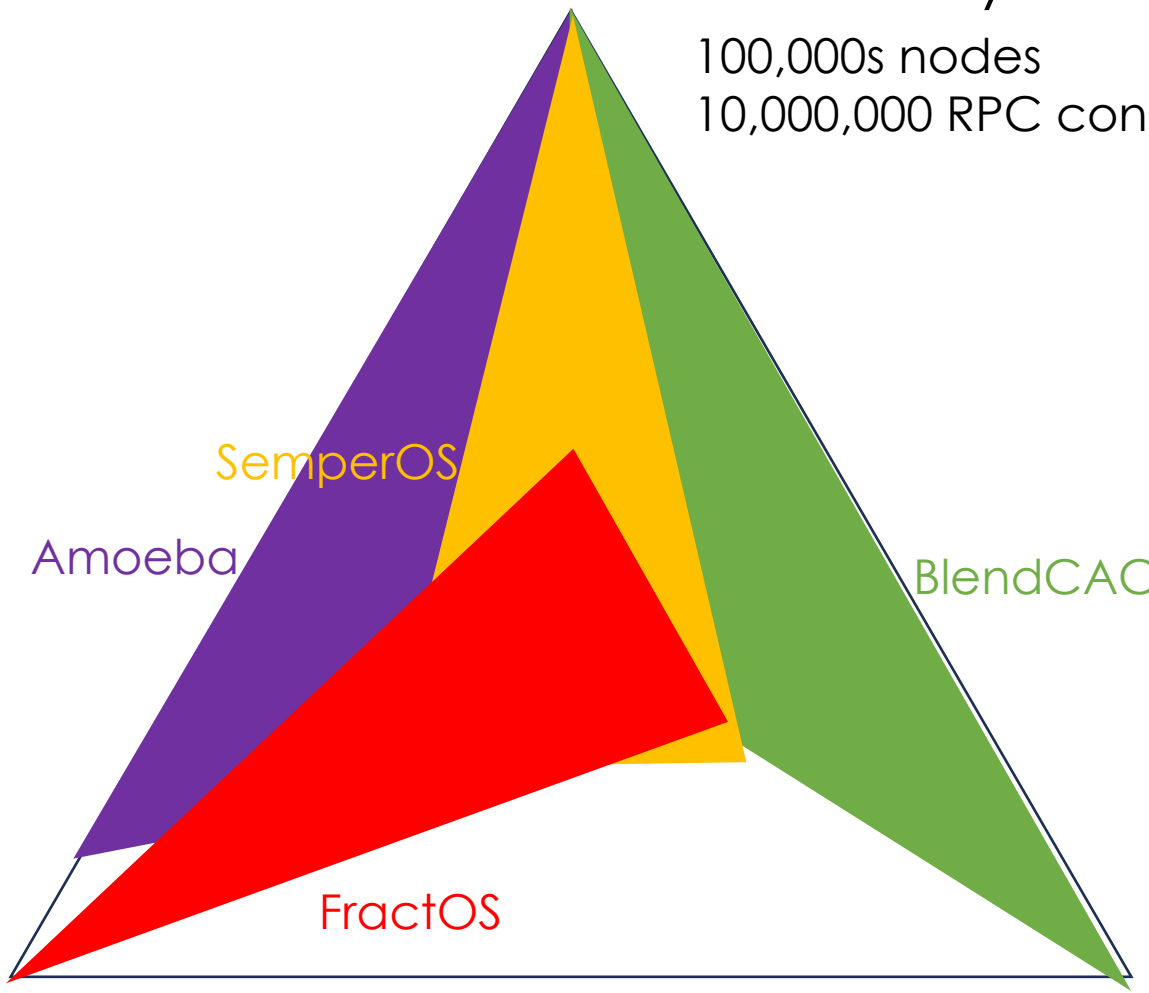
No Capability Systems Are Cloud Ready



No Capability Systems Are Cloud Ready

Scalability

100,000s nodes
10,000,000 RPC connections/sec



Critical-path Performance

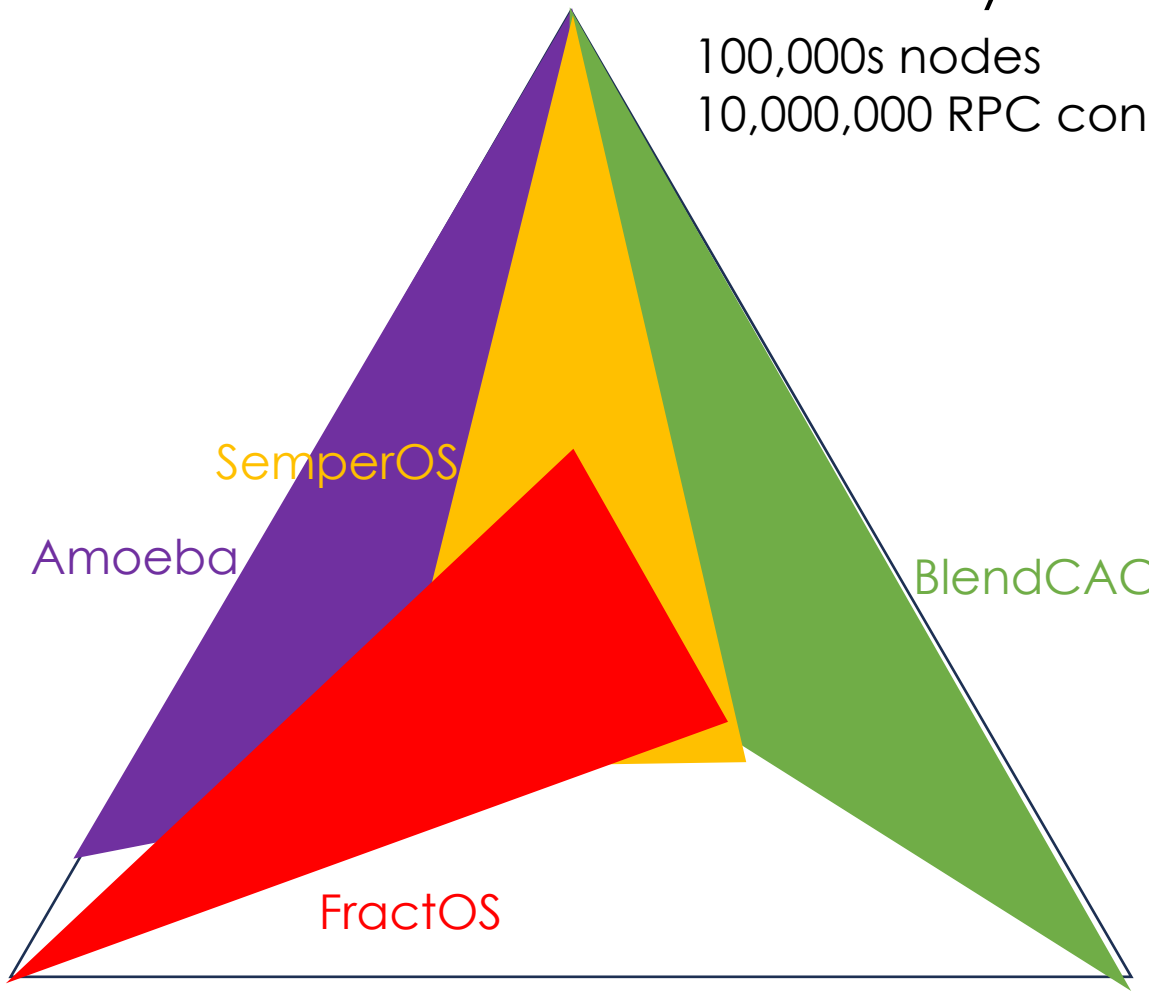
1-2 µs-scale RPCs

Fault-tolerance

<17.5 failures/node day
~35,000 failures on 2k nodes

No Capability Systems Are Cloud Ready

We need all three to achieve cloud scale…



Scalability

100,000s nodes
10,000,000 RPC connections/sec

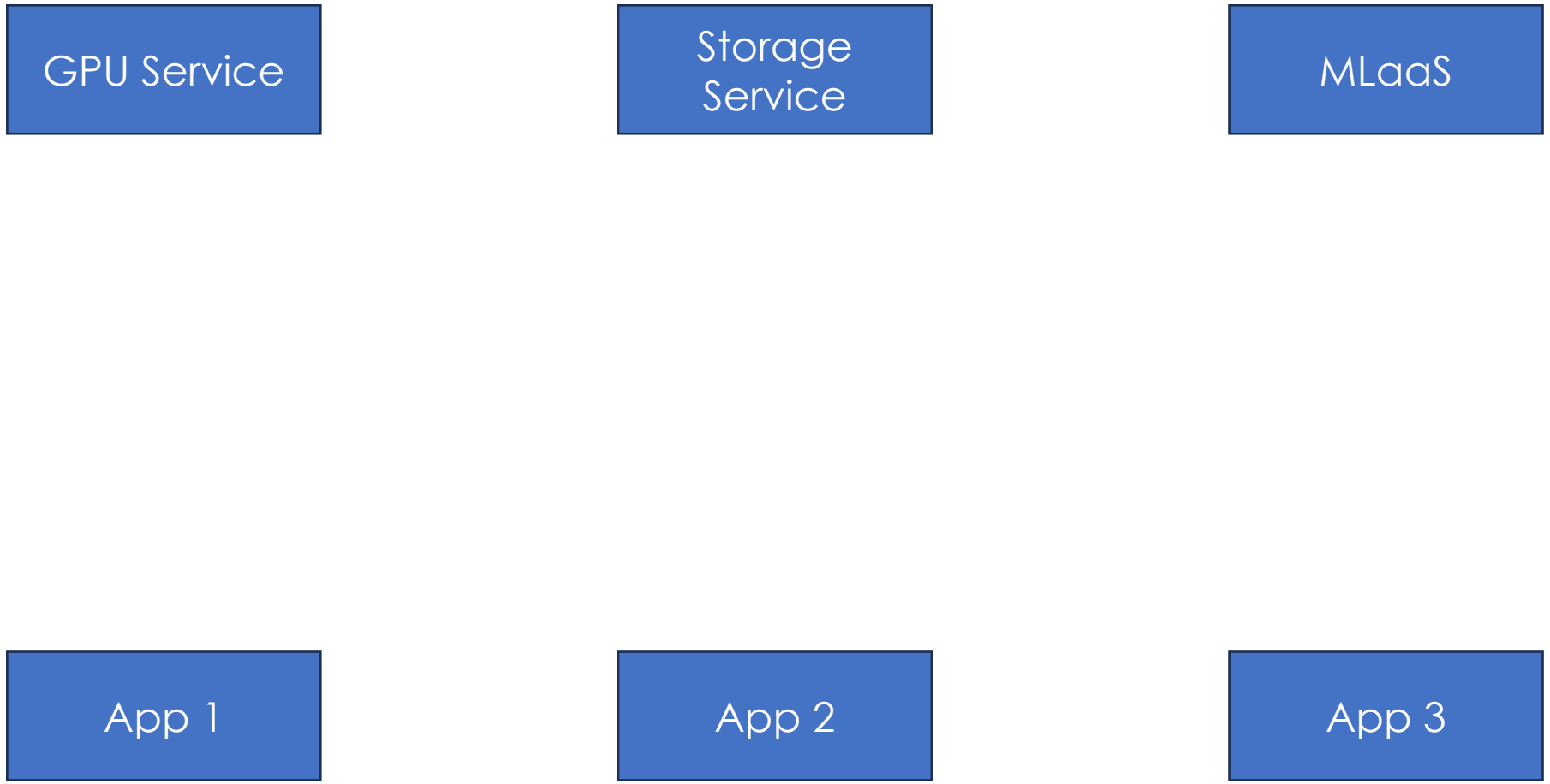
Critical-path Performance

1-2 µs-scale RPCs

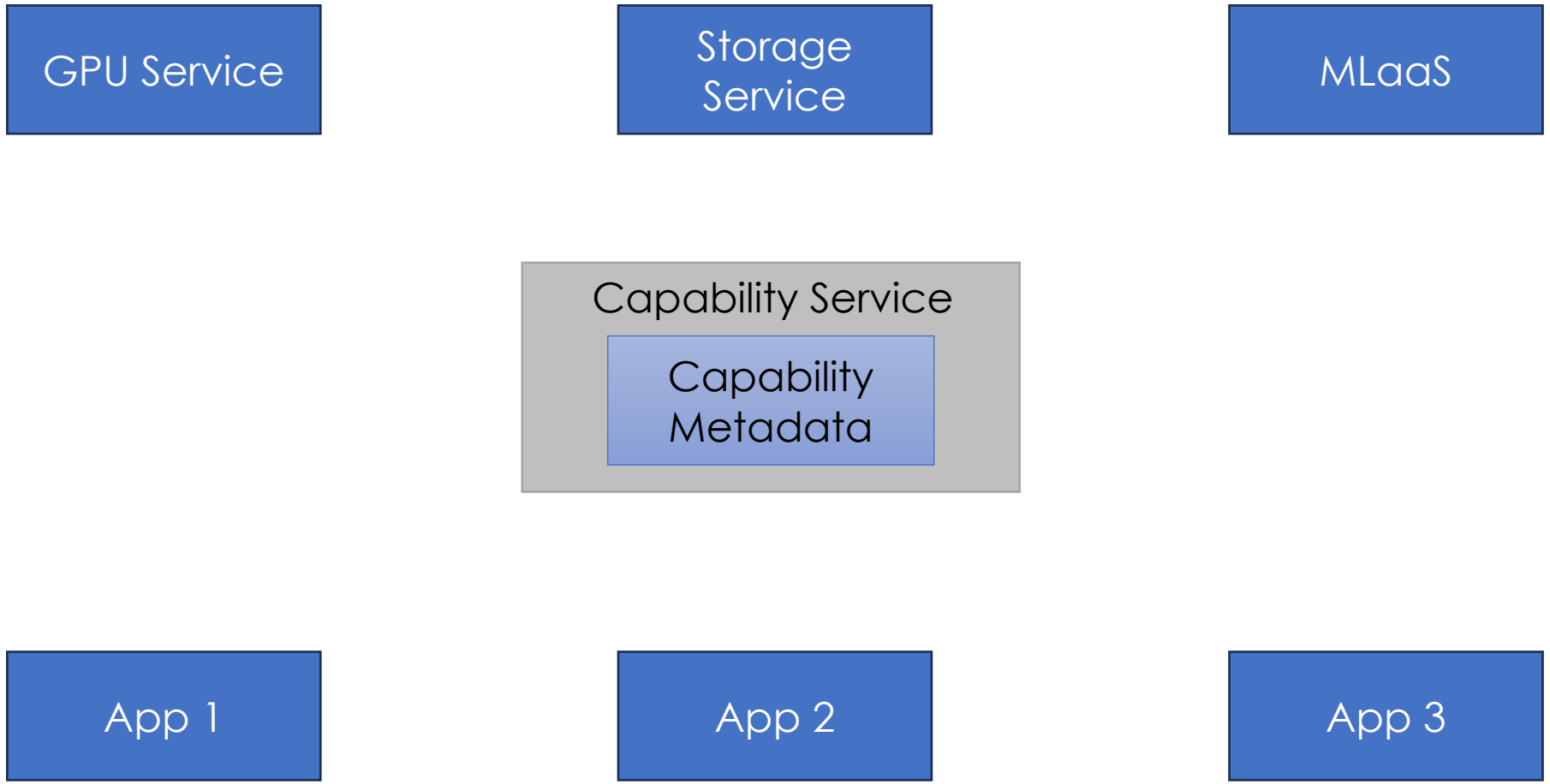
Fault-tolerance

<17.5 failures/node day
~35,000 failures on 2k nodes

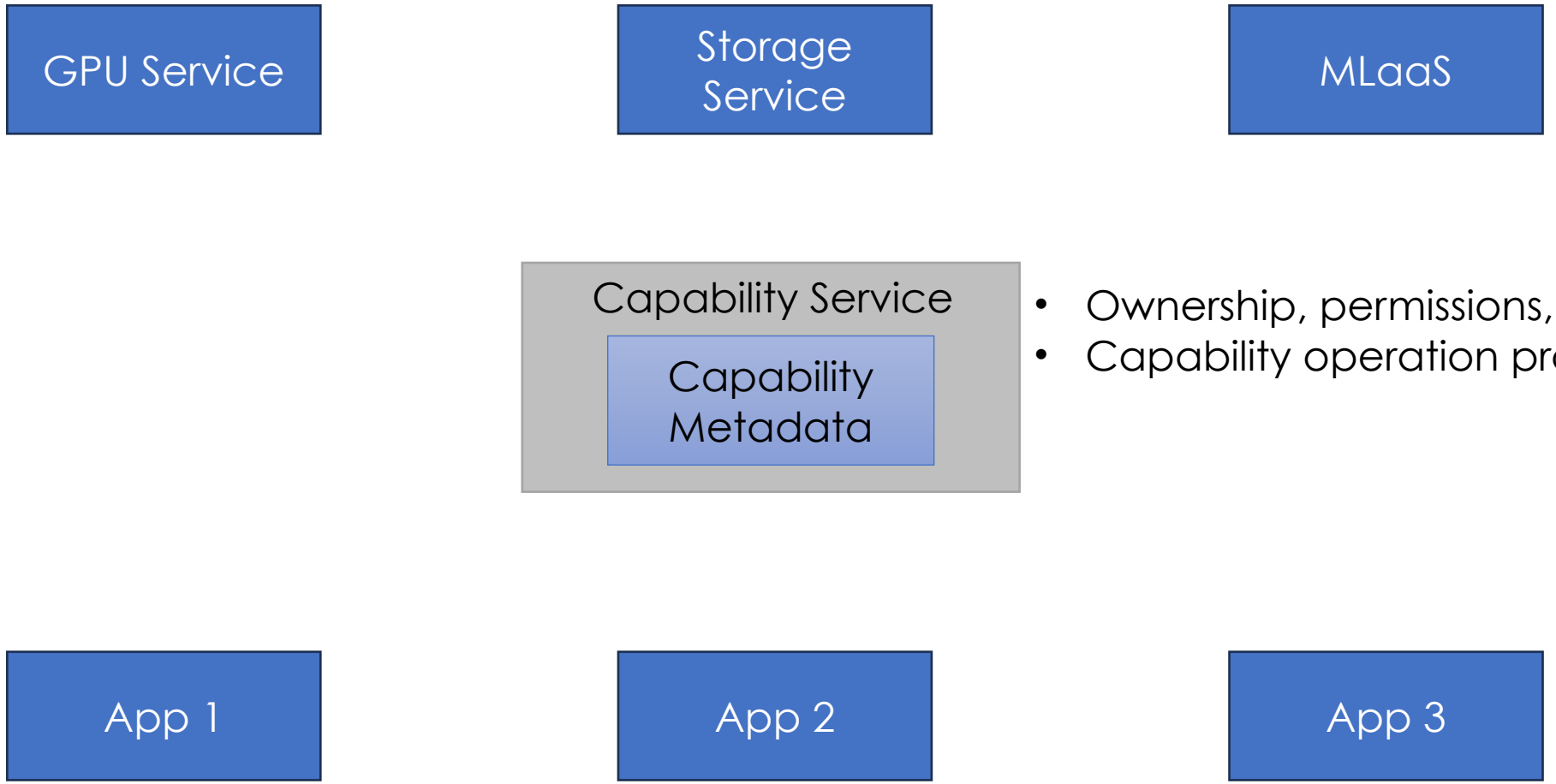
Problem: Centralized Metadata in Existing Capability Systems



Problem: Centralized Metadata in Existing Capability Systems

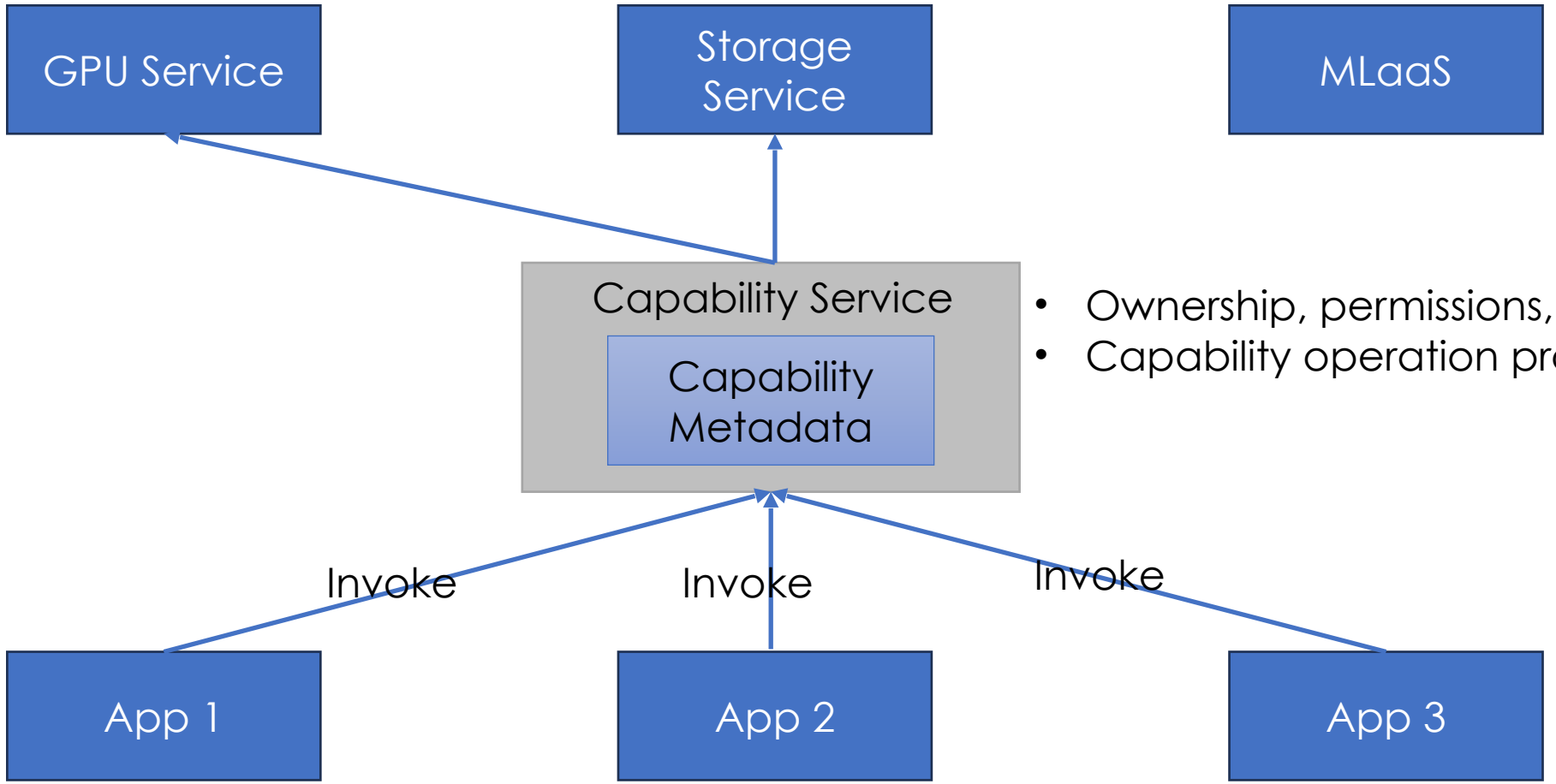


Problem: Centralized Metadata in Existing Capability Systems



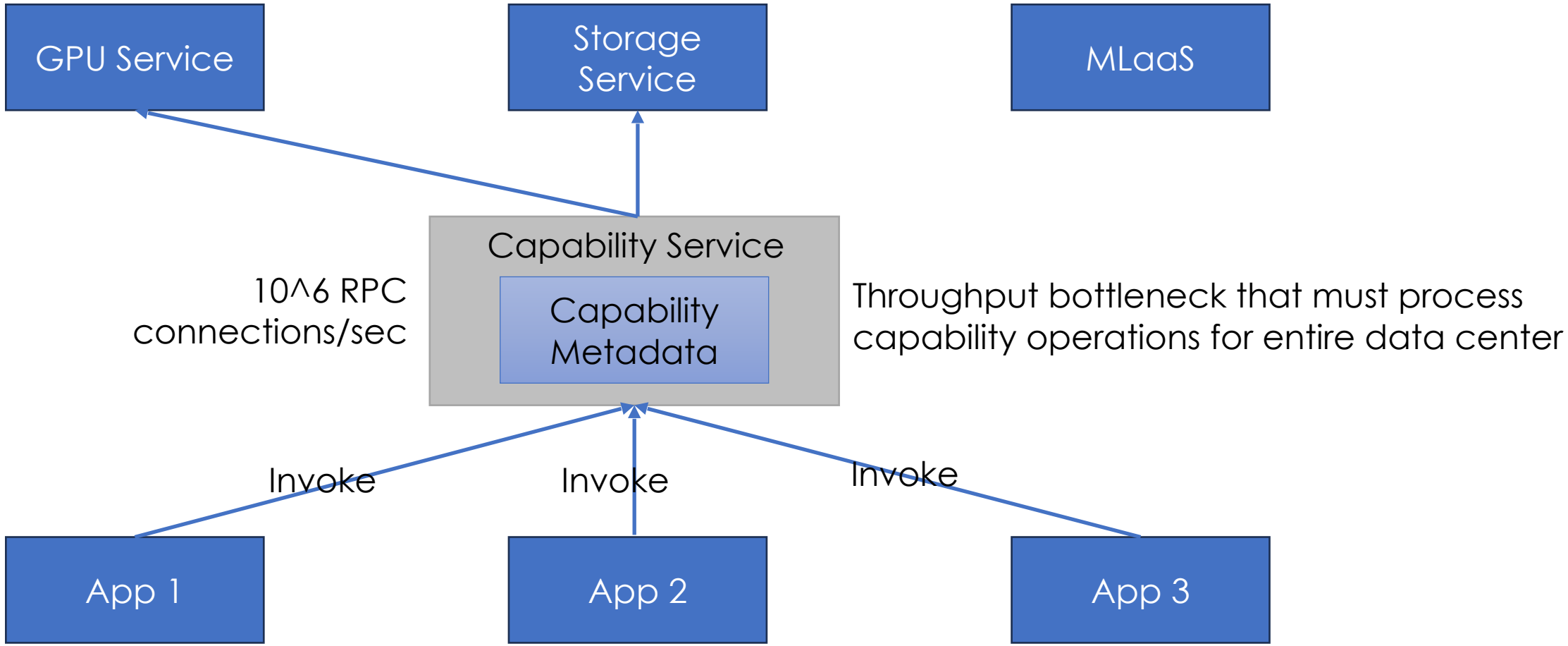
- Ownership, permissions, …
- Capability operation processing

Problem: Centralized Metadata in Existing Capability Systems

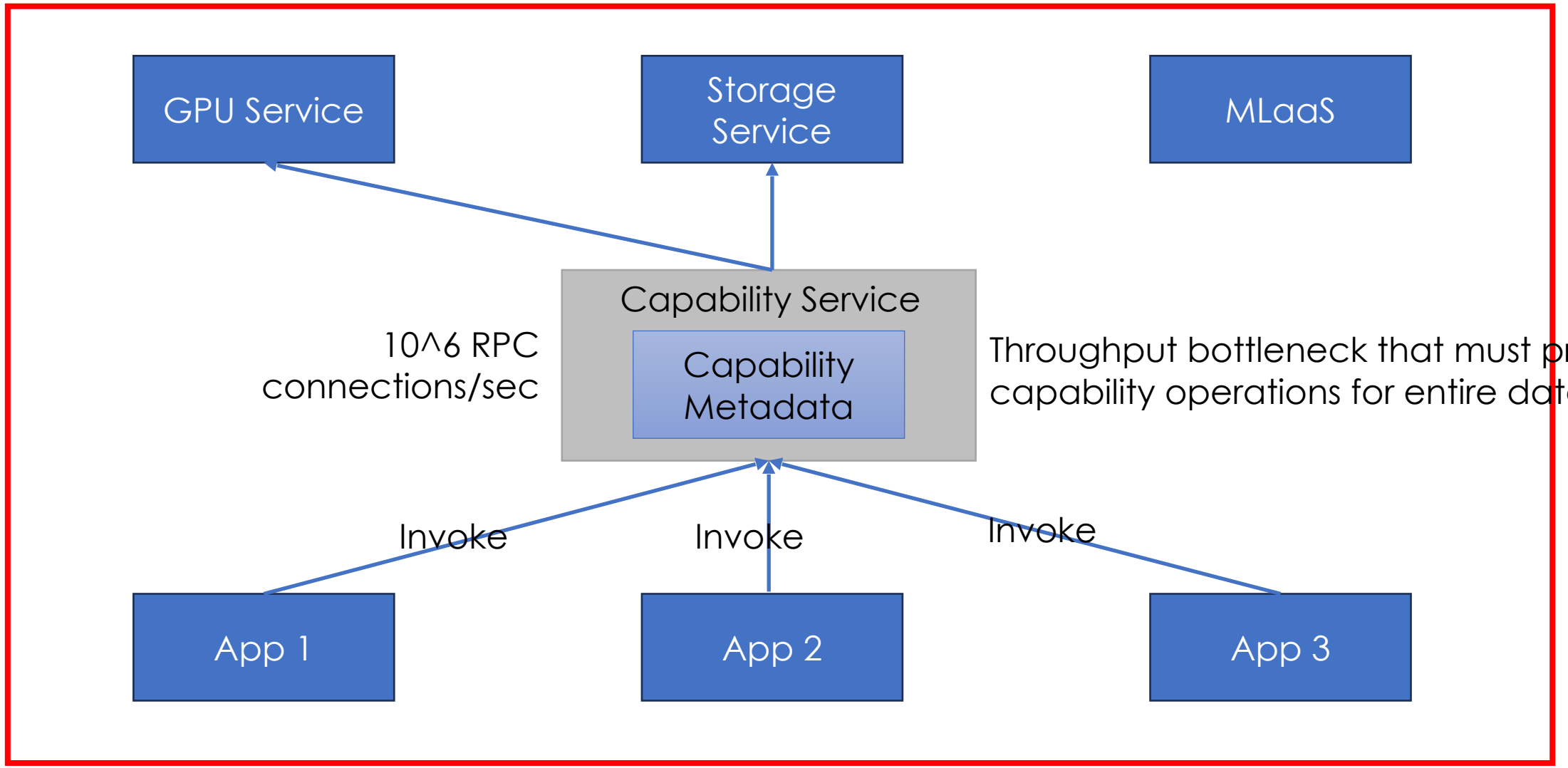


- Ownership, permissions, …
- Capability operation processing

Problem: Centralized Metadata in Existing Capability Systems

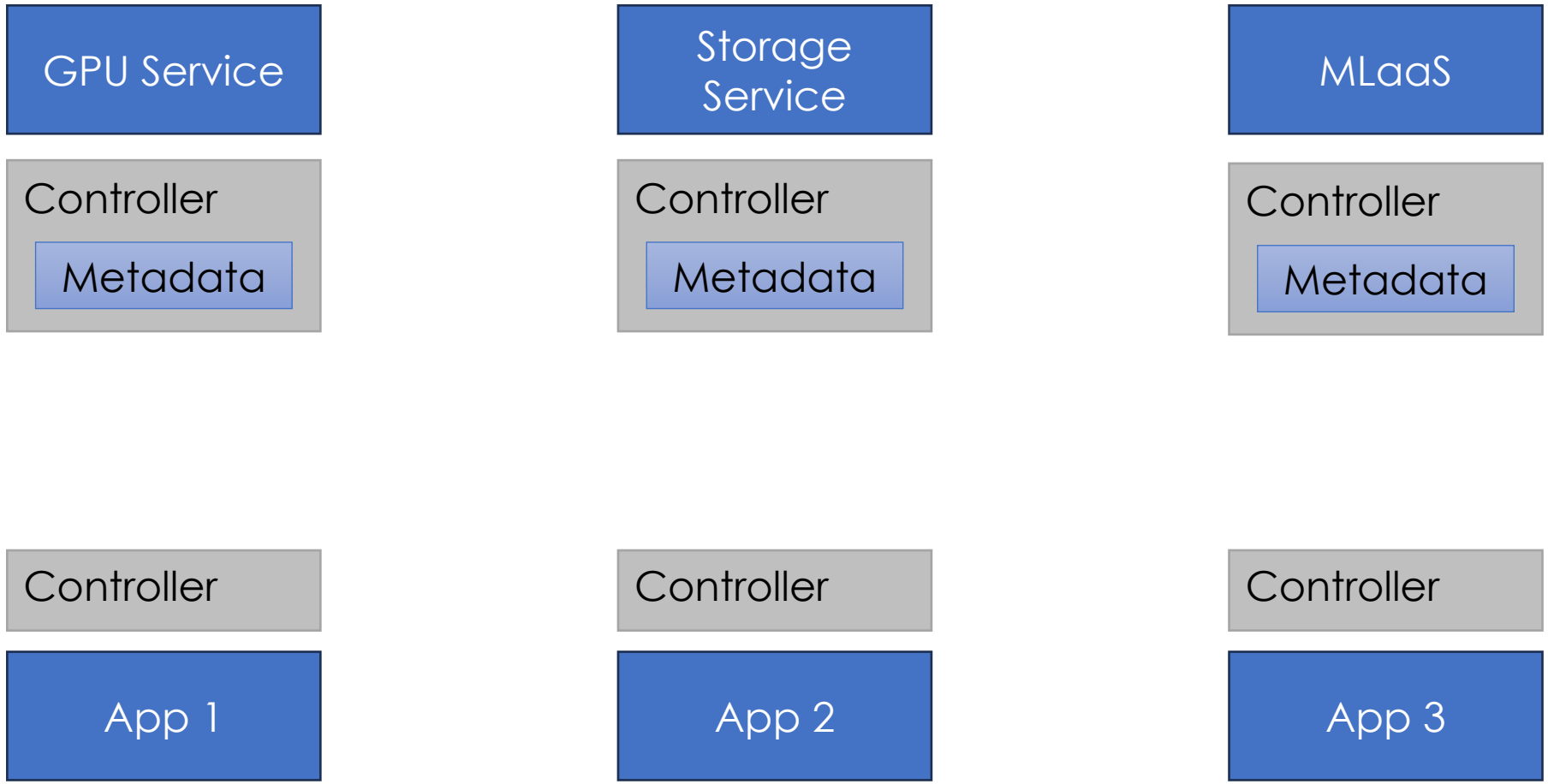


Problem: Centralized Metadata in Existing Capability Systems

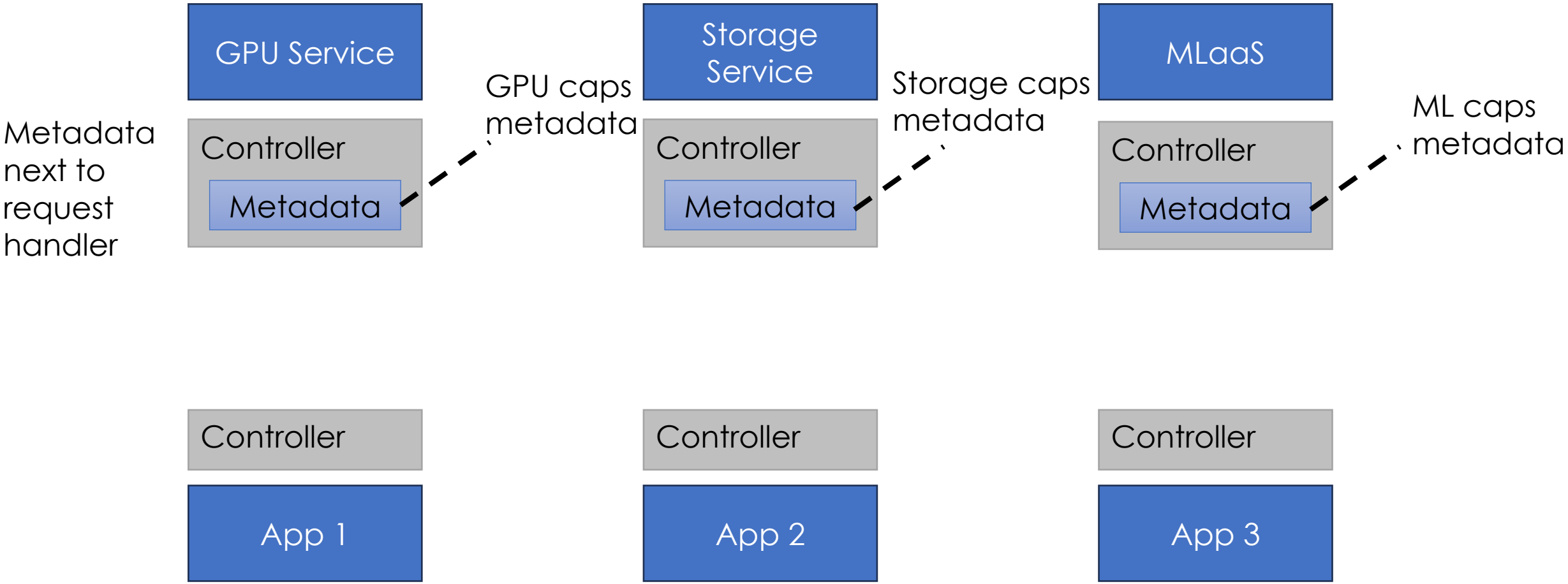


Single point of failure – if the capability service fails the system is no longer usable

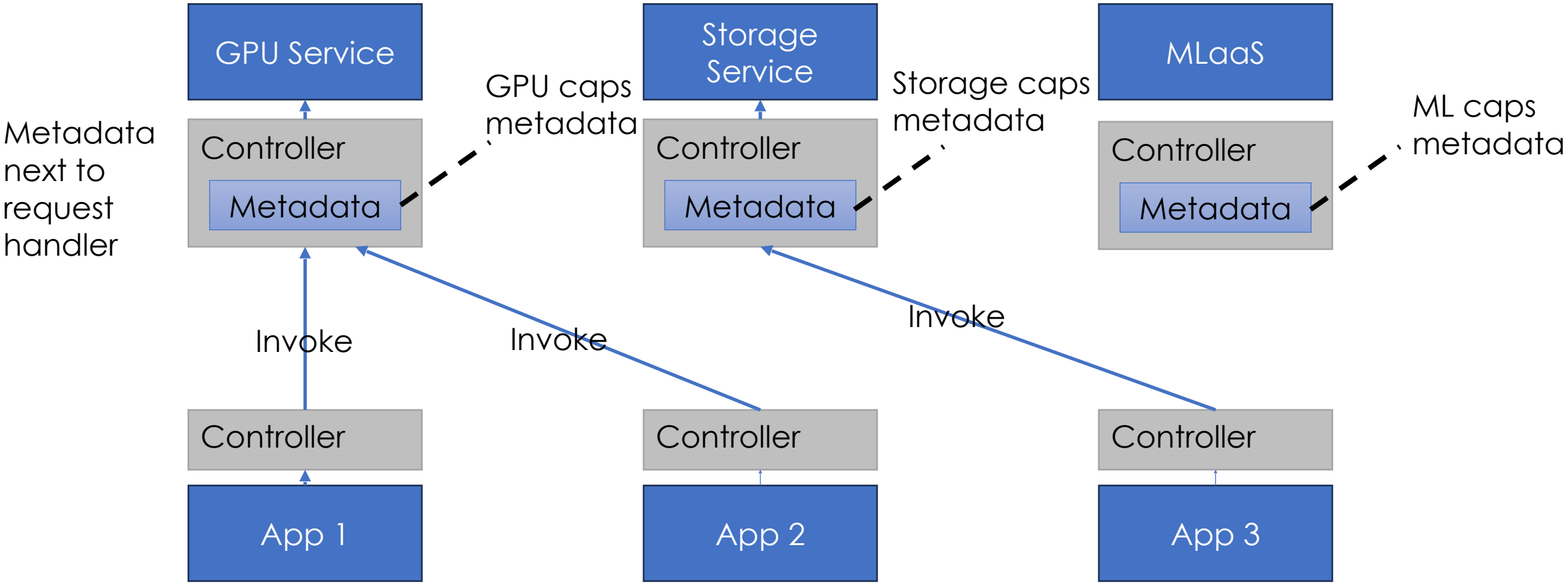
Owner-based Metadata Sharding



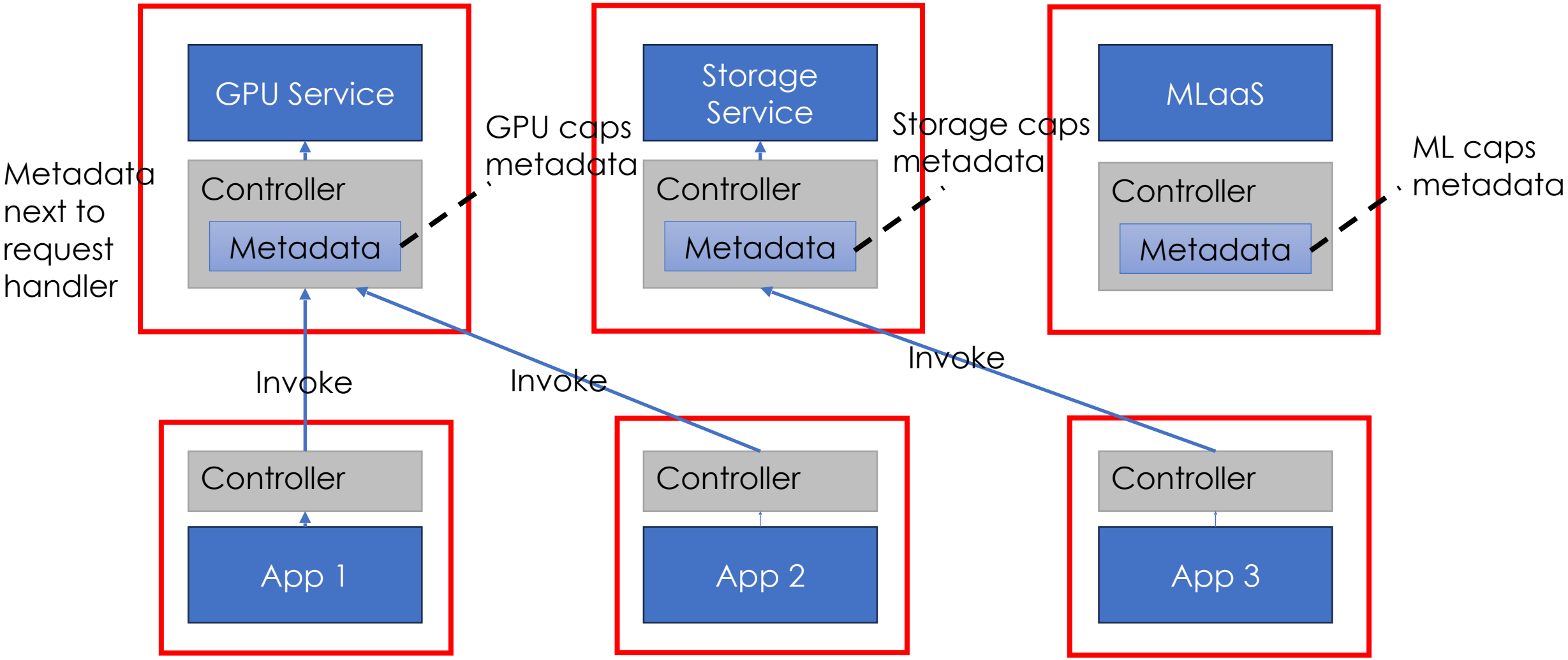
Owner-based Metadata Sharding



Owner-based Metadata Sharding

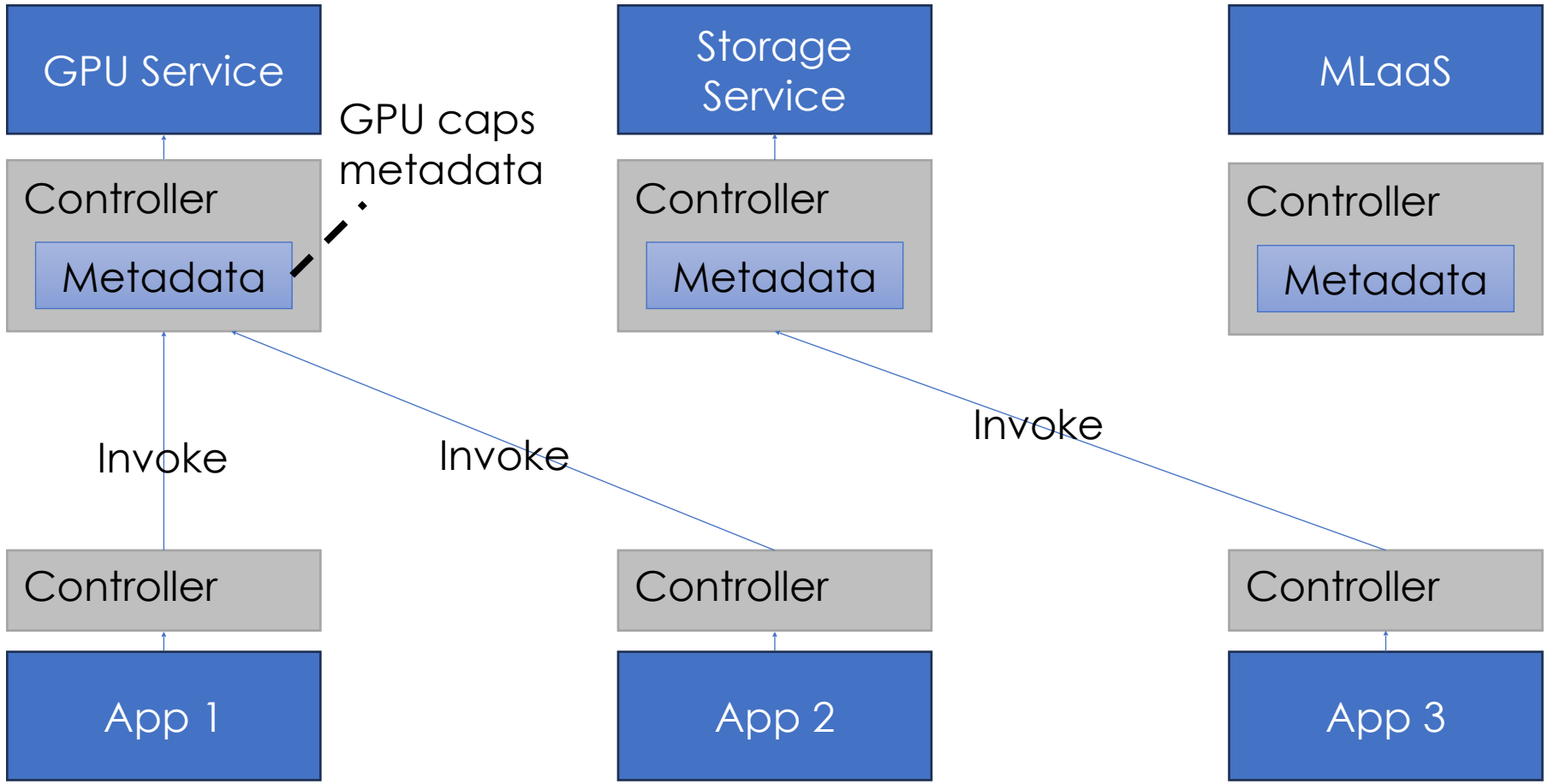


Owner-based Metadata Sharding

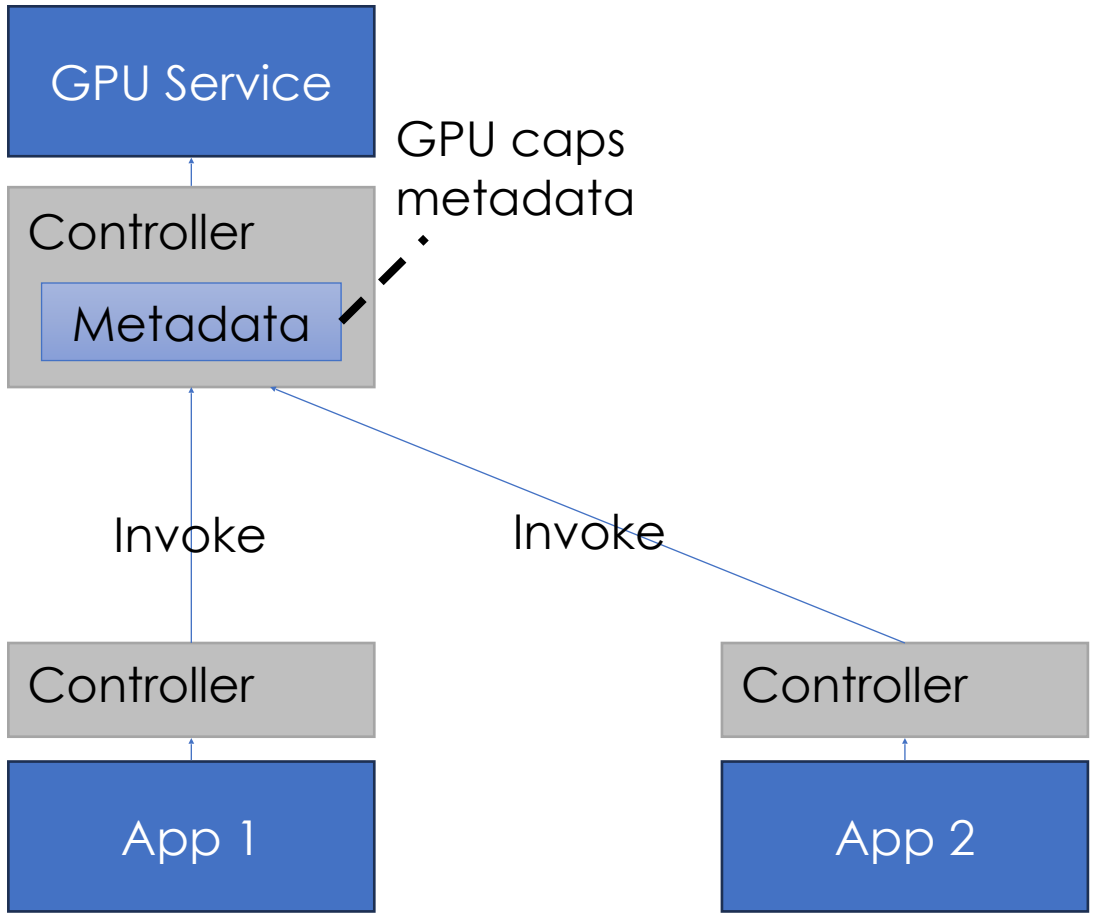


No throughput bottleneck, isolated failure domains

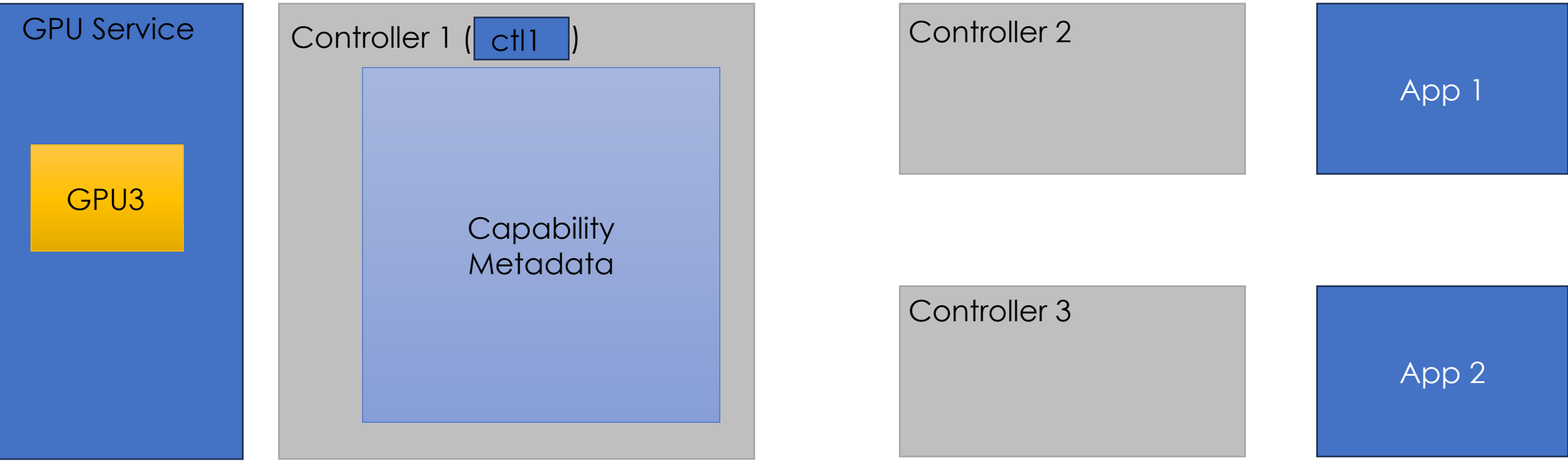
Problem: Inefficient Revocation in Existing Capability Systems



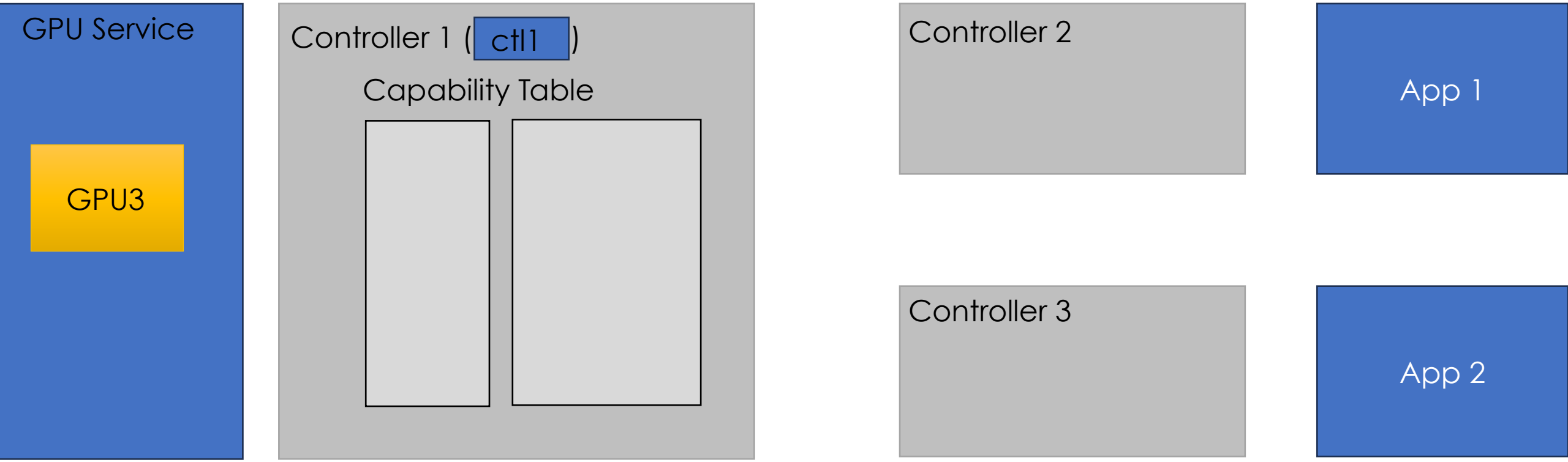
Problem: Inefficient Revocation in Existing Capability Systems



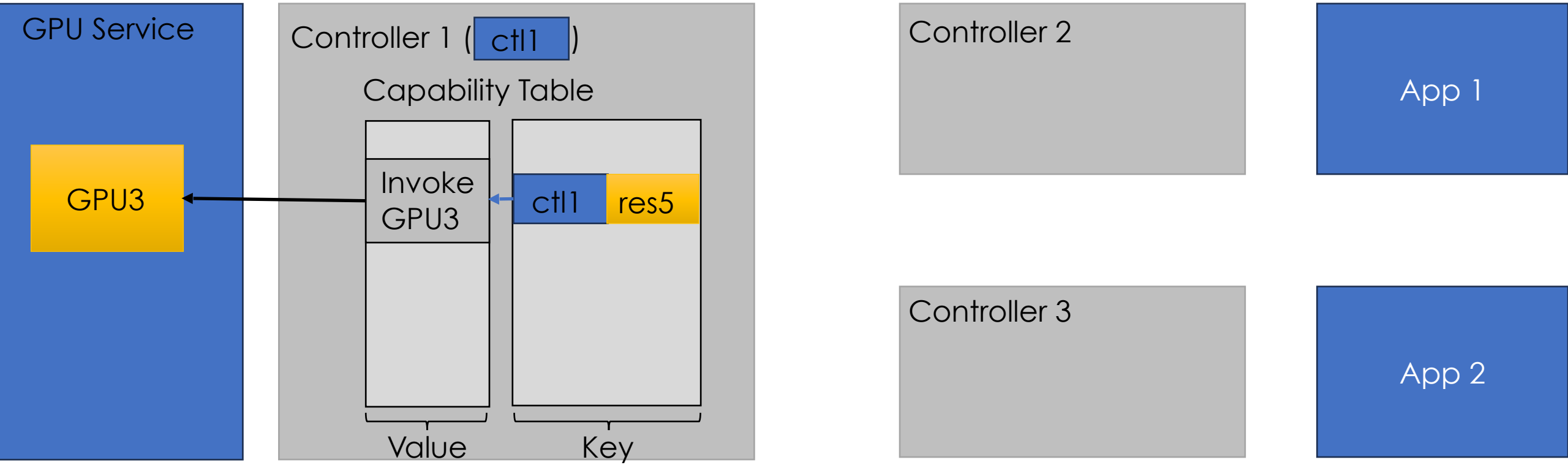
Problem: Inefficient Revocation in Existing Capability Systems



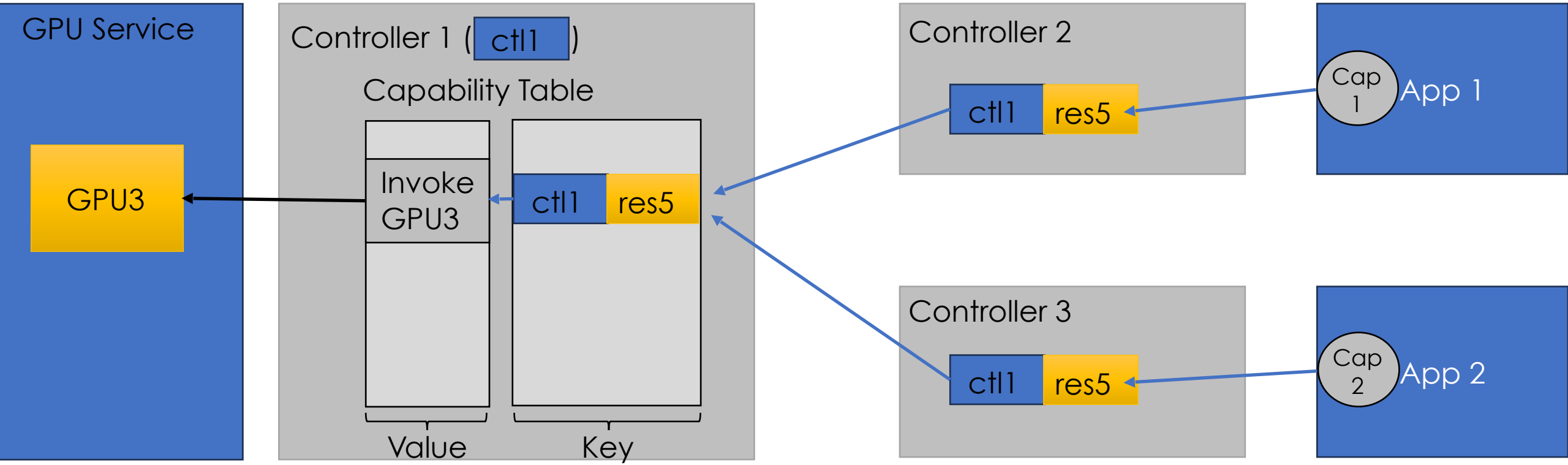
Problem: Inefficient Revocation in Existing Capability Systems



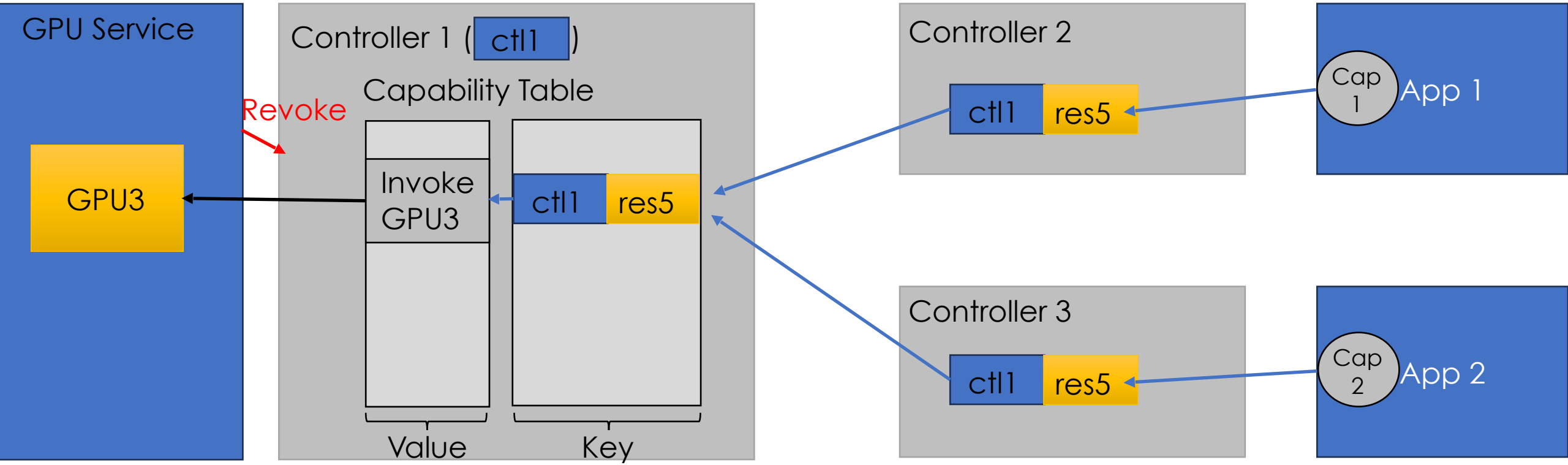
Problem: Inefficient Revocation in Existing Capability Systems



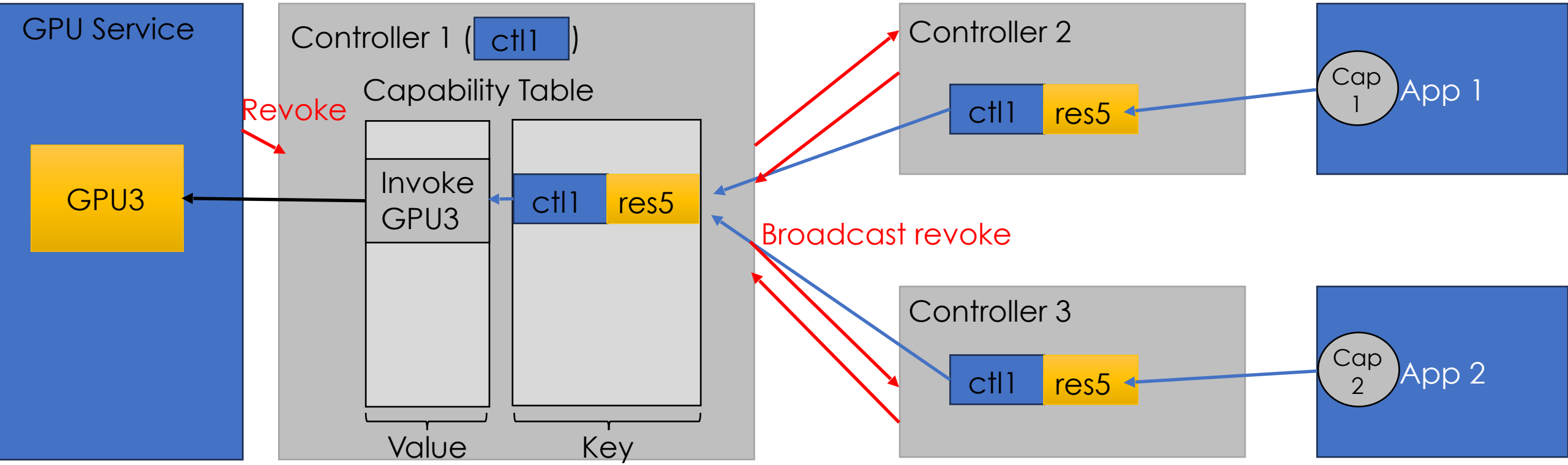
Problem: Inefficient Revocation in Existing Capability Systems



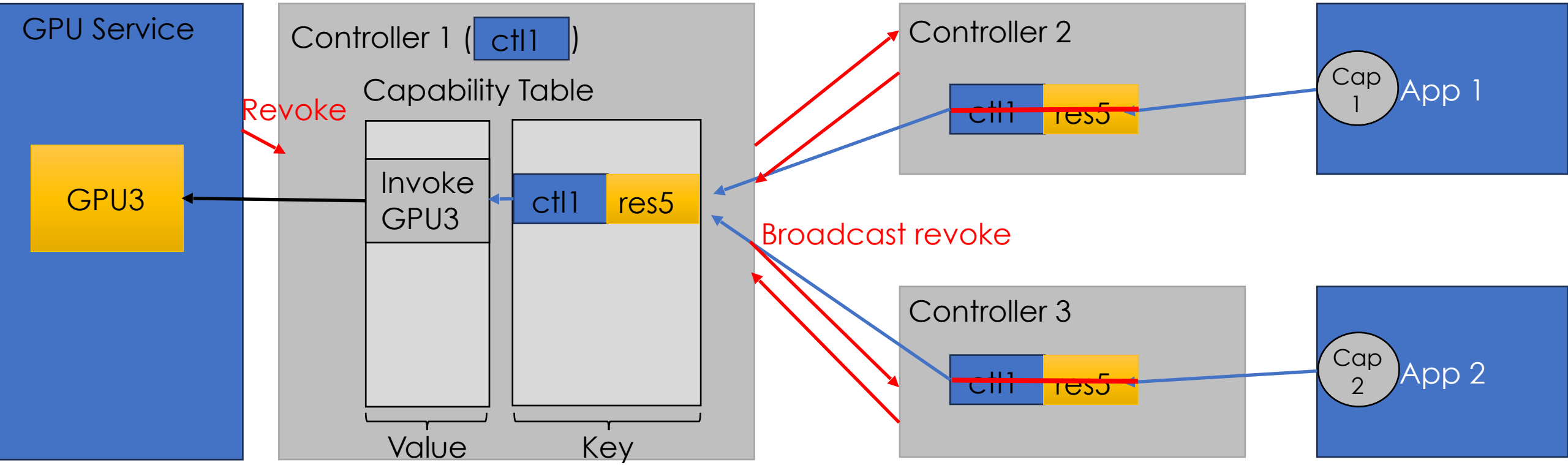
Problem: Inefficient Revocation in Existing Capability Systems



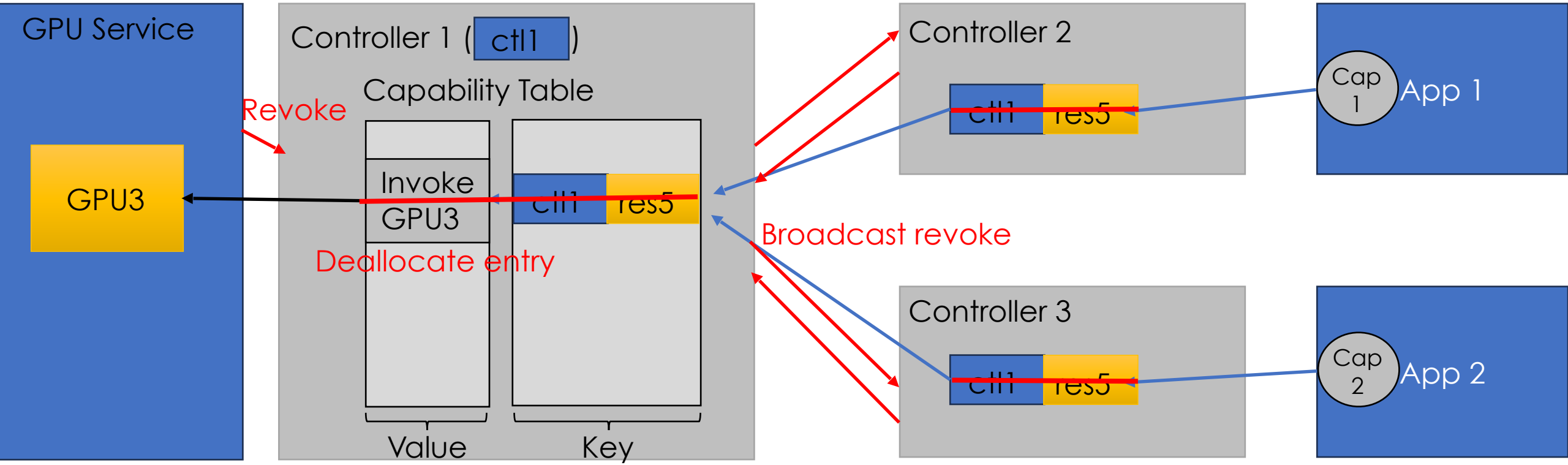
Problem: Inefficient Revocation in Existing Capability Systems



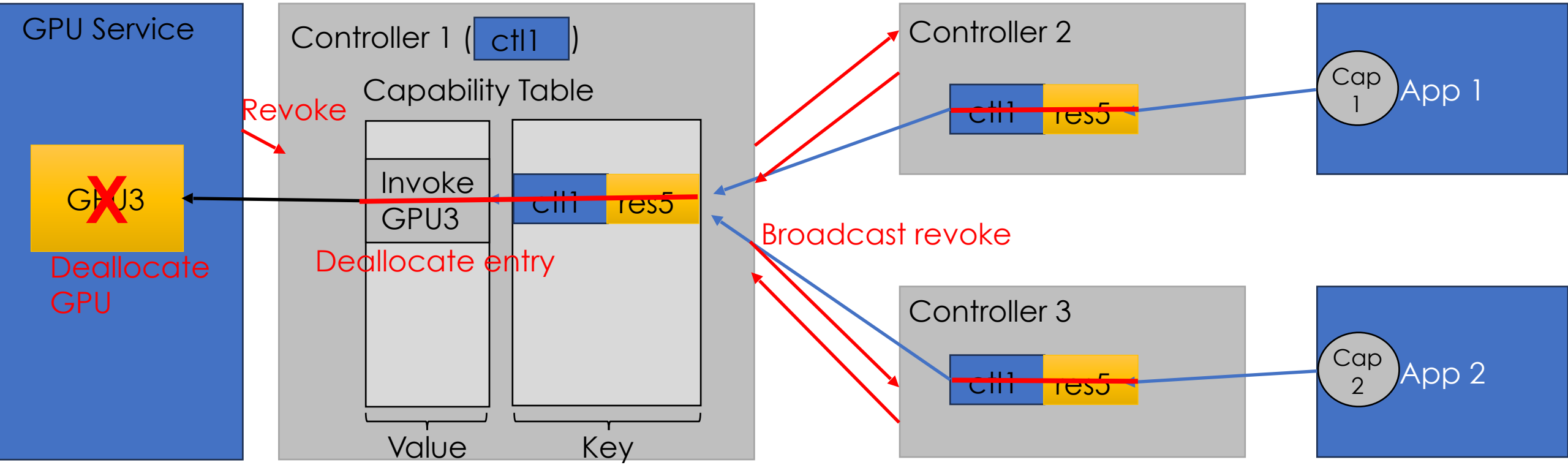
Problem: Inefficient Revocation in Existing Capability Systems



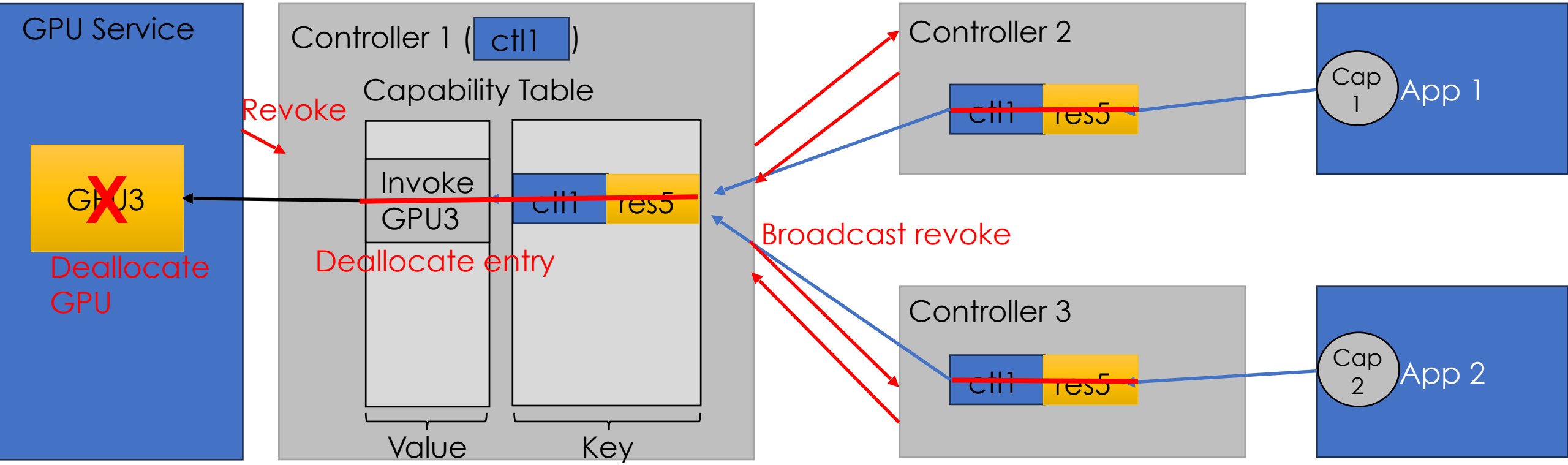
Problem: Inefficient Revocation in Existing Capability Systems



Problem: Inefficient Revocation in Existing Capability Systems

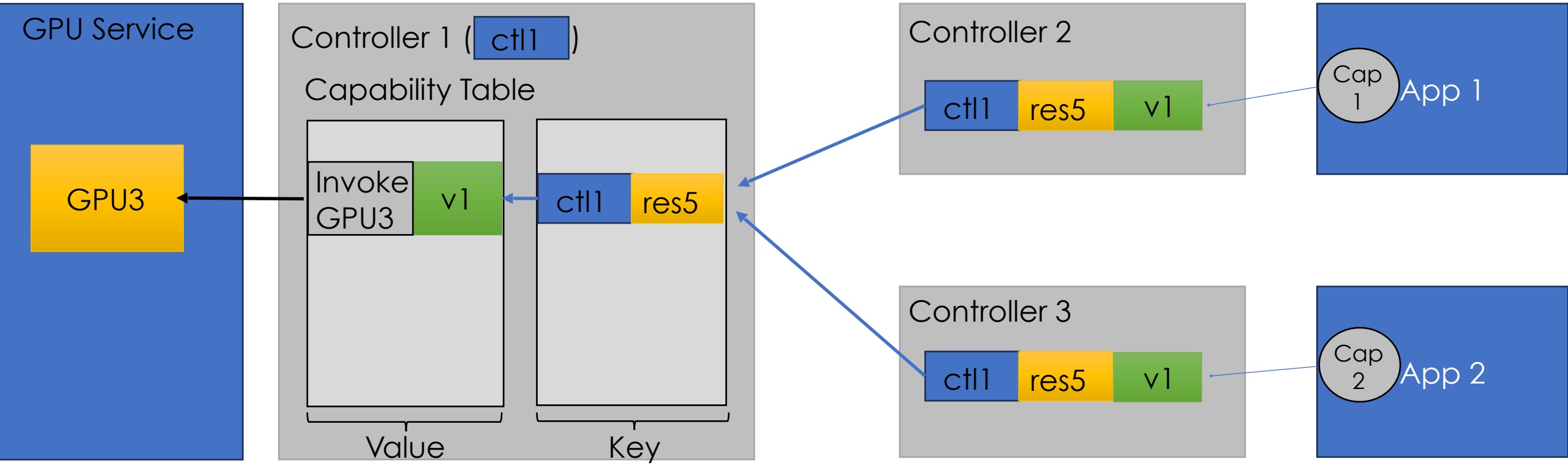


Problem: Inefficient Revocation in Existing Capability Systems

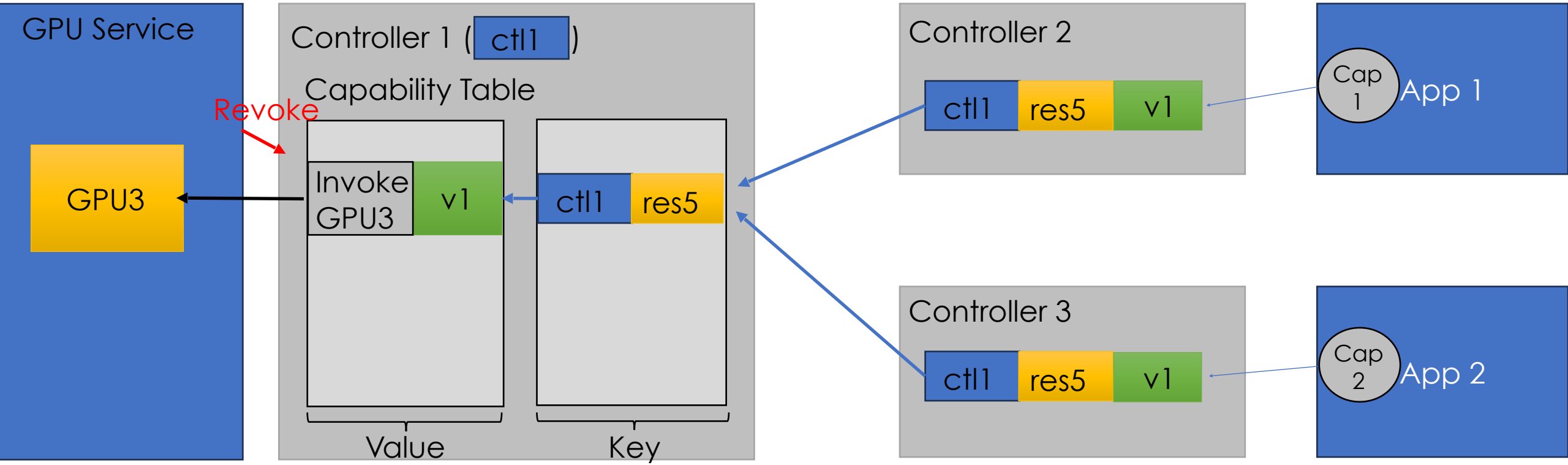


Revocation requires O(N) network messages (and CPU)

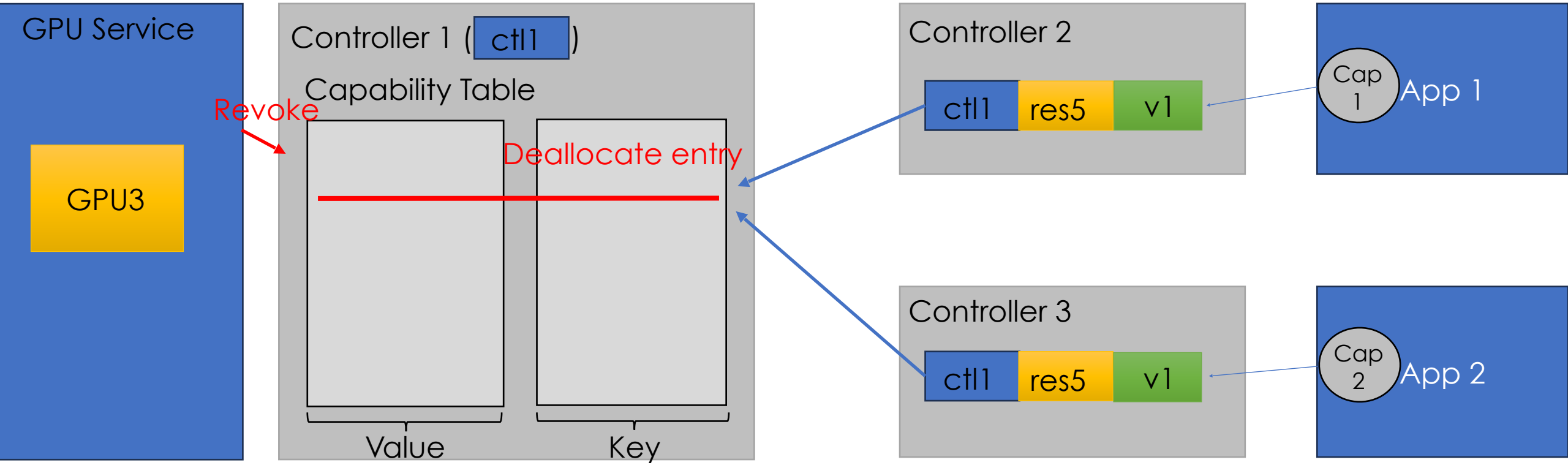
Versioned Capabilities



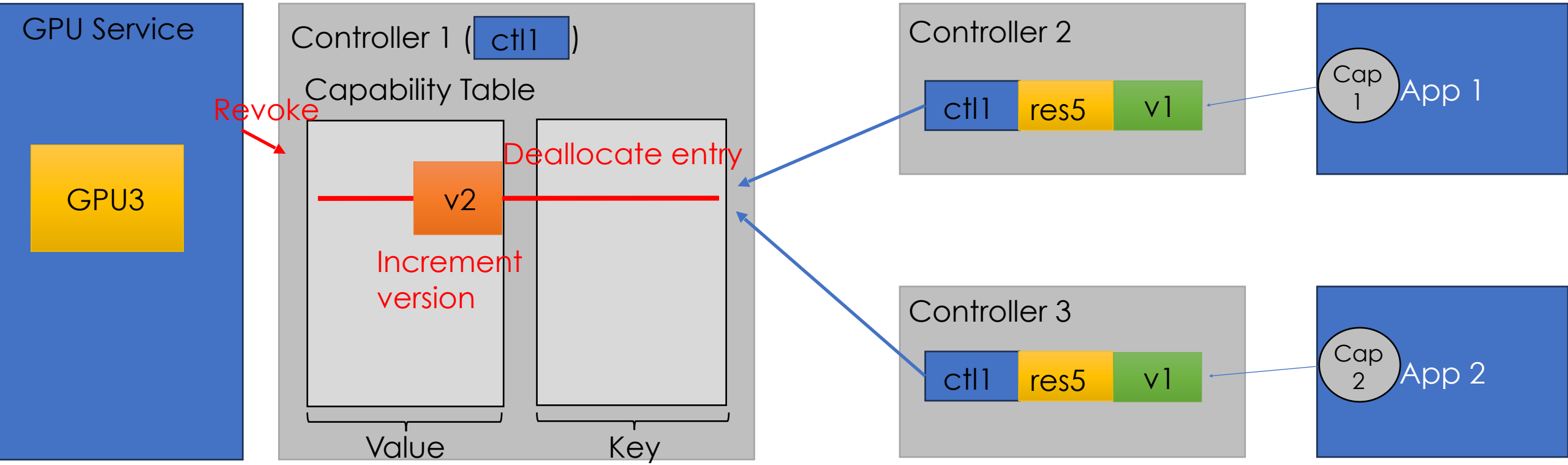
Versioned Capabilities



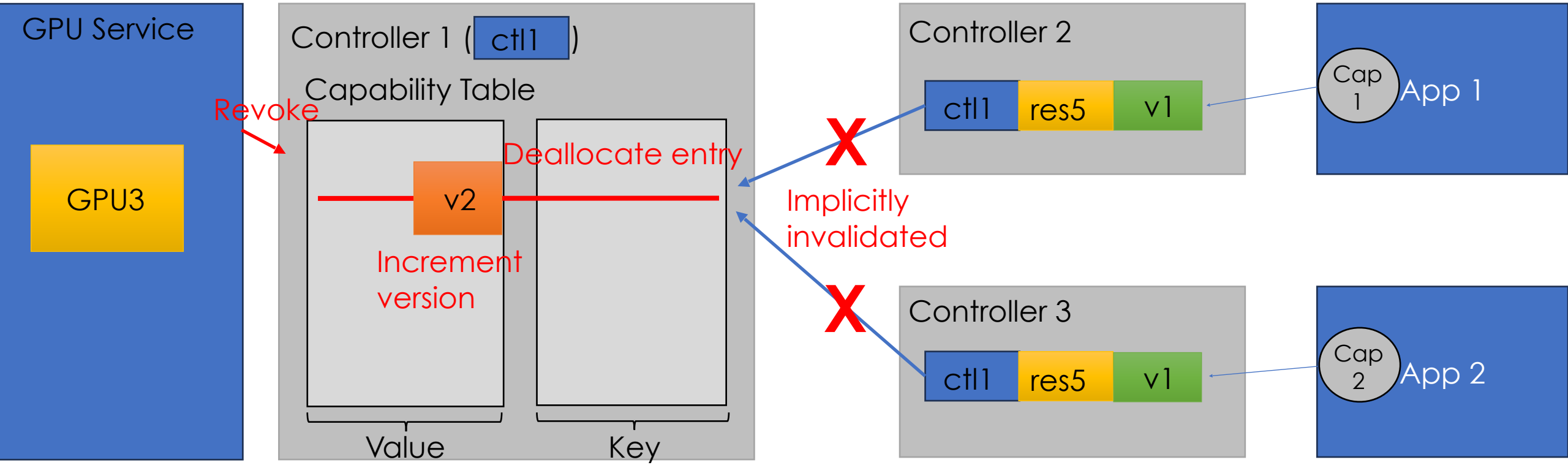
Versioned Capabilities



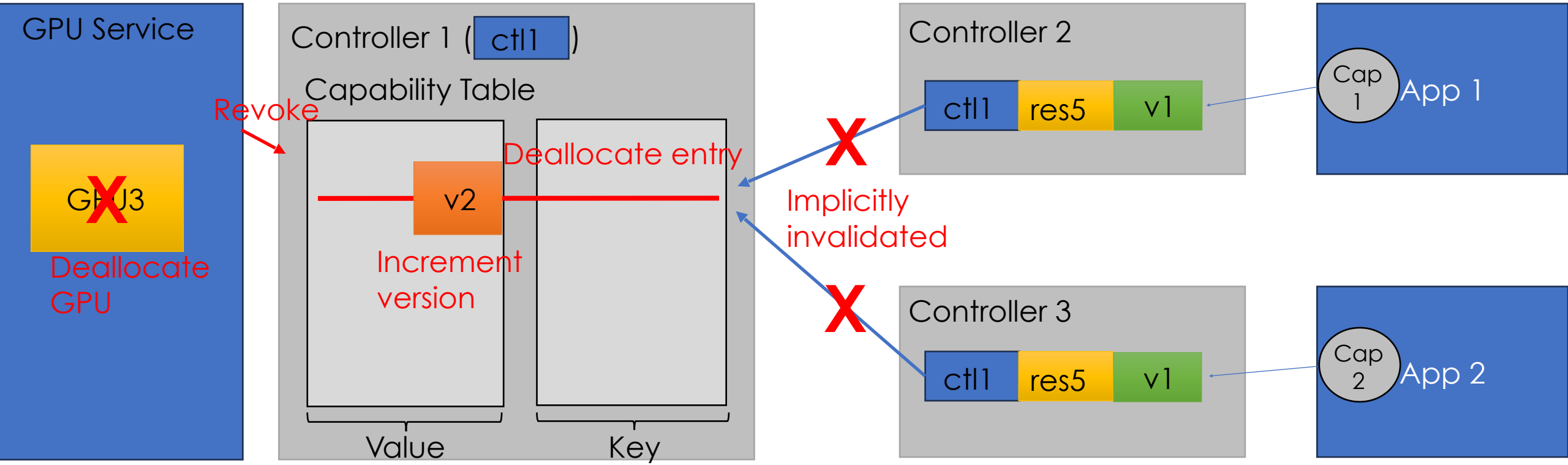
Versioned Capabilities



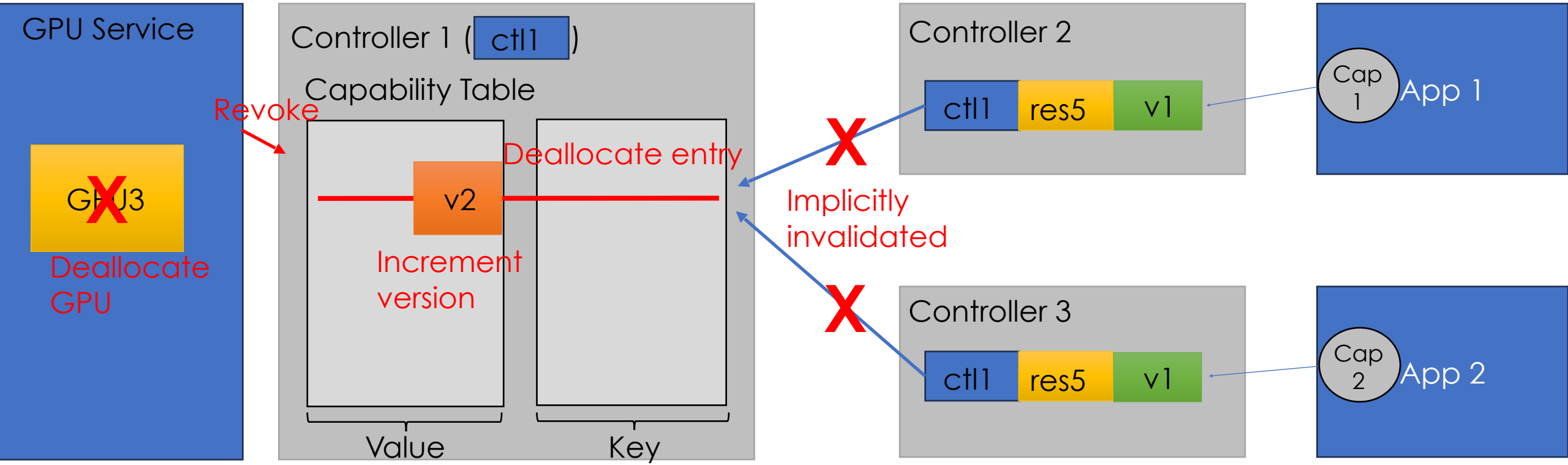
Versioned Capabilities



Versioned Capabilities

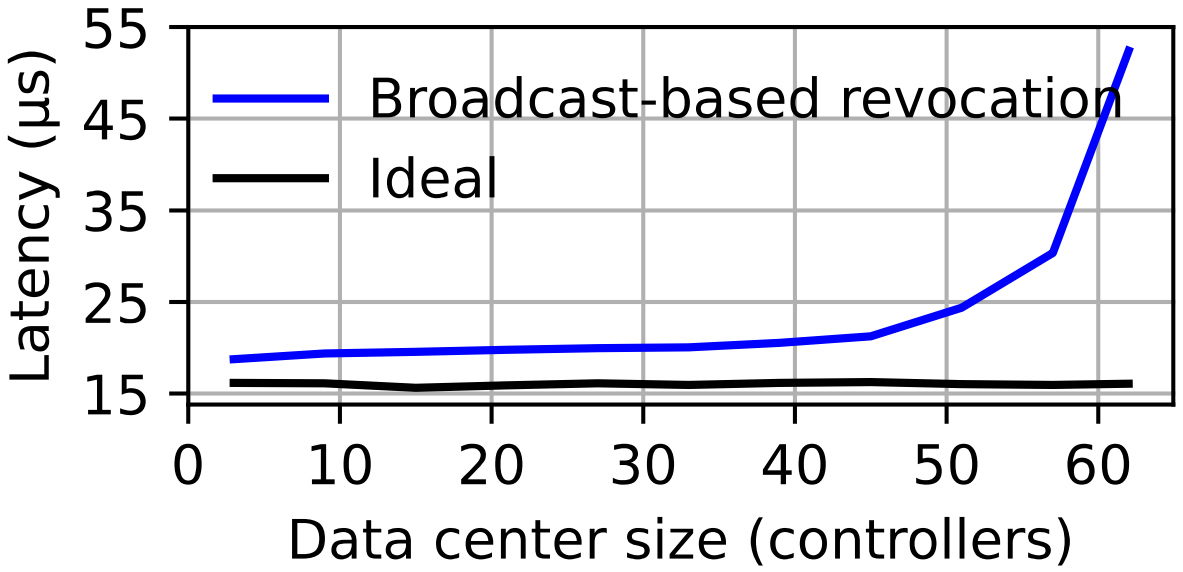


Versioned Capabilities

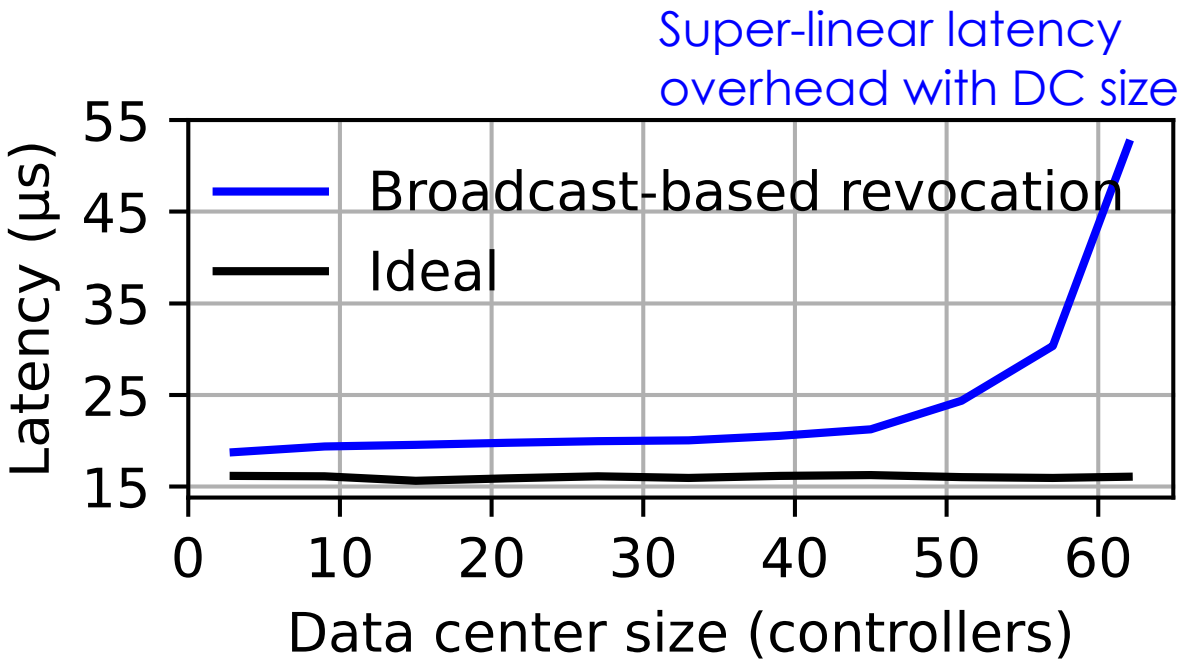


Revocation requires zero network transmission (or CPU)

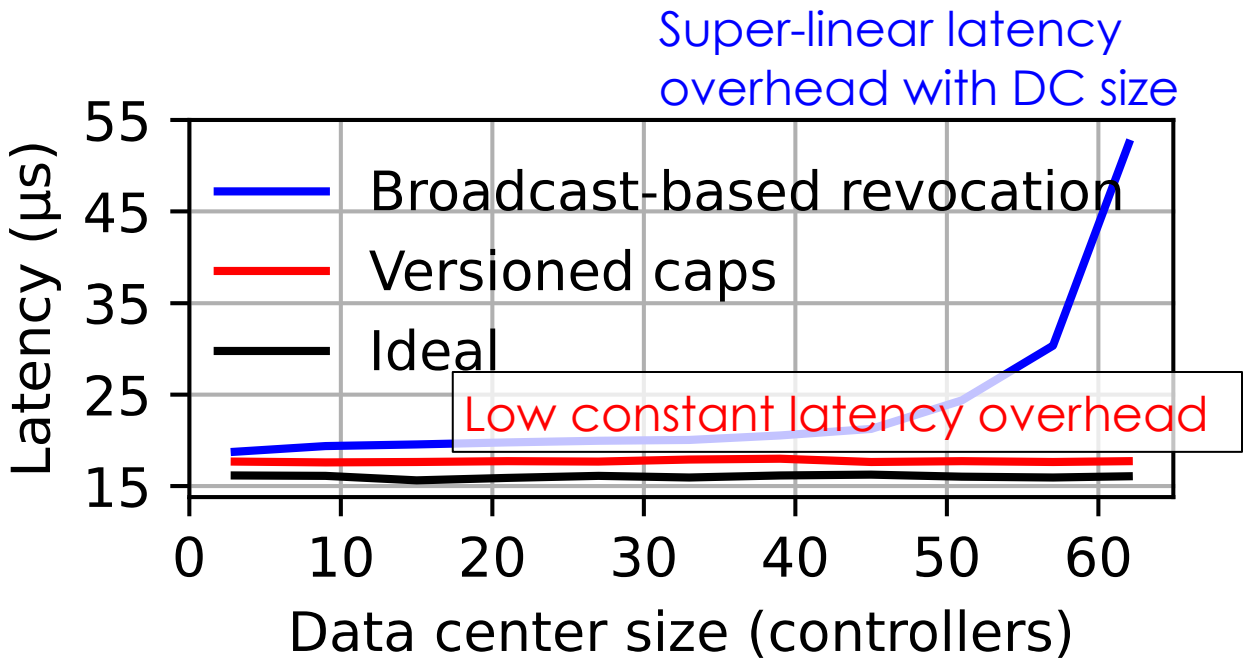
Evaluation


Latency (µs)
55
45
35
25
15
0
10
20
30
40
50
60
Data center size (controllers)
Broadcast-based revocation
Ideal

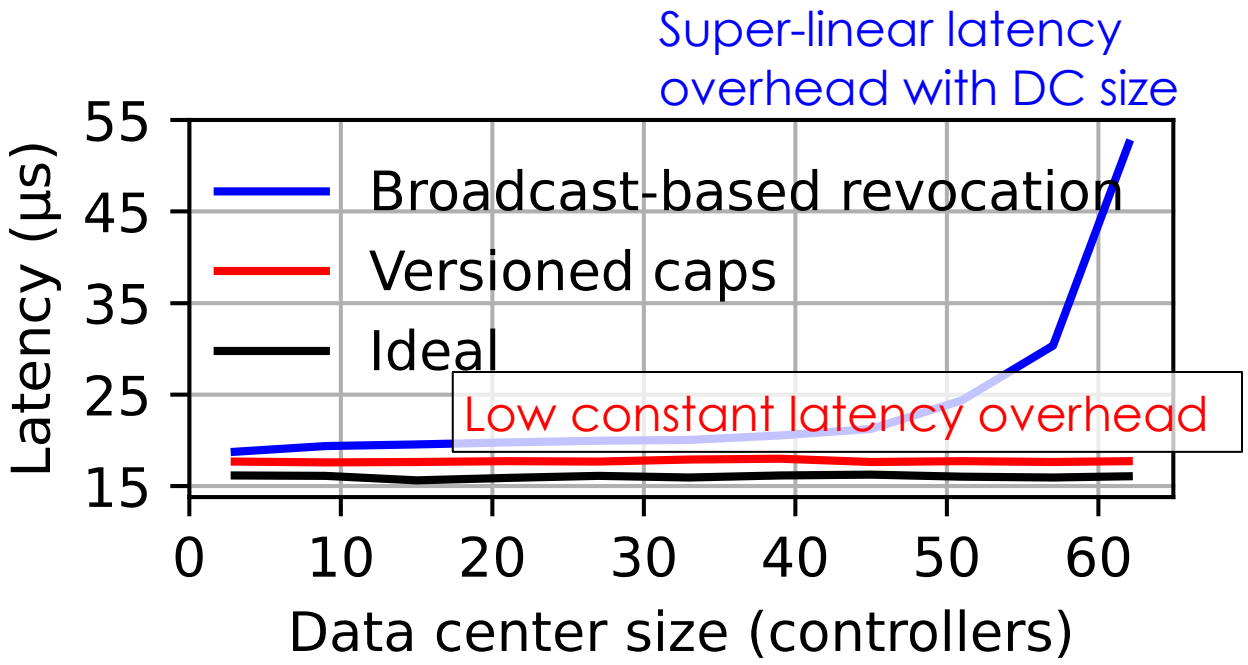
Evaluation

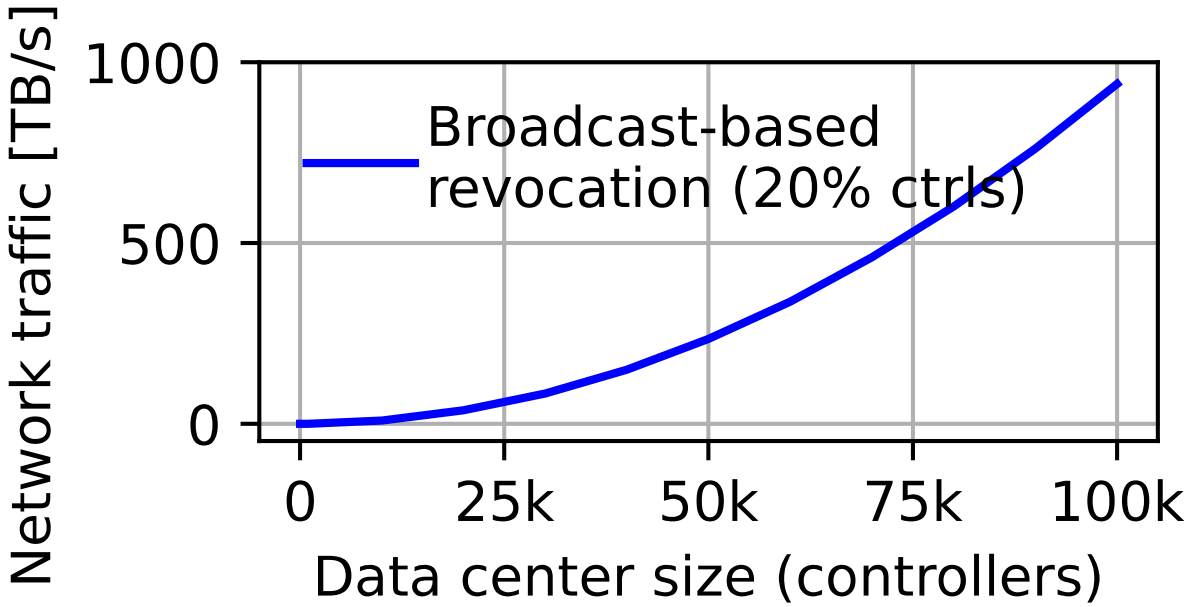

Super-linear latency
overhead with DC size
Latency (µs)
55
45
35
25
15
Broadcast-based revocation
Ideal
0
10
20
30
40
50
60
Data center size (controllers)

Evaluation

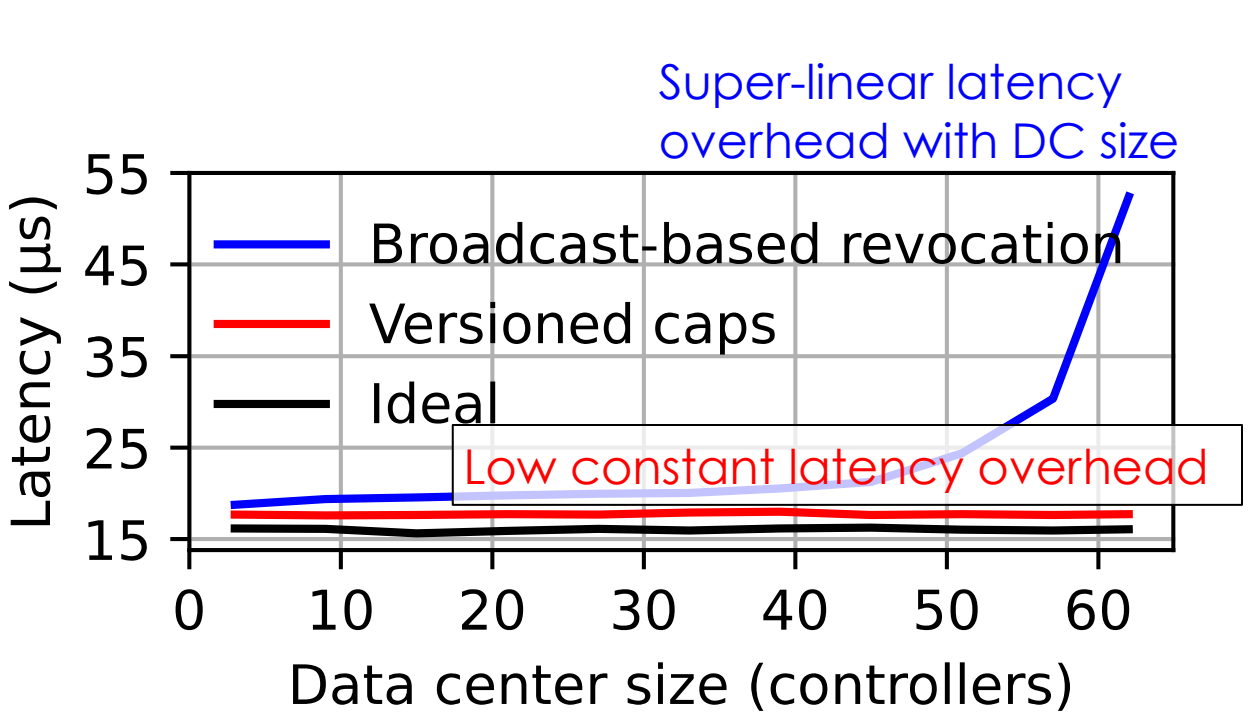

Super-linear latency
overhead with DC size
Broadcast-based revocation
Versioned caps
Ideal
Low constant latency overhead
Latency (µs)
55
45
35
25
15
0
10
20
30
40
50
60
Data center size (controllers)

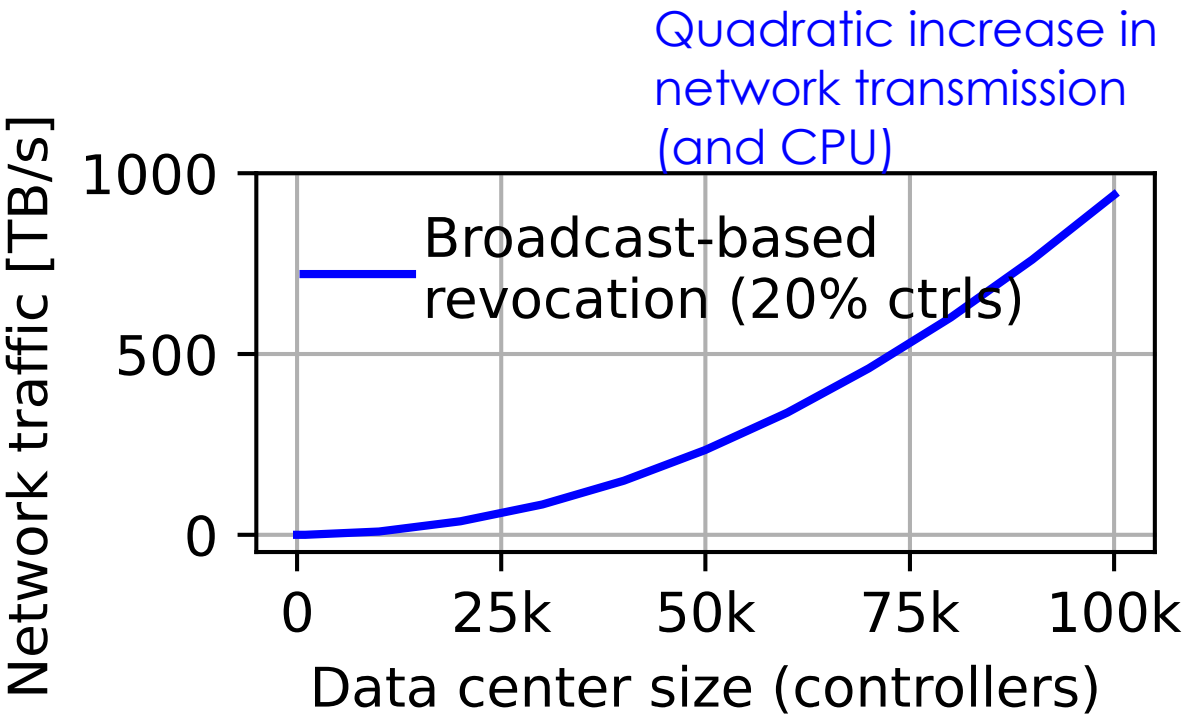
Evaluation


Super-linear latency
overhead with DC size
Latency (µs)
55
45
35
25
15
Broadcast-based revocation
Versioned caps
Ideal
Low constant latency overhead
0
10
20
30
40
50
60
Data center size (controllers)

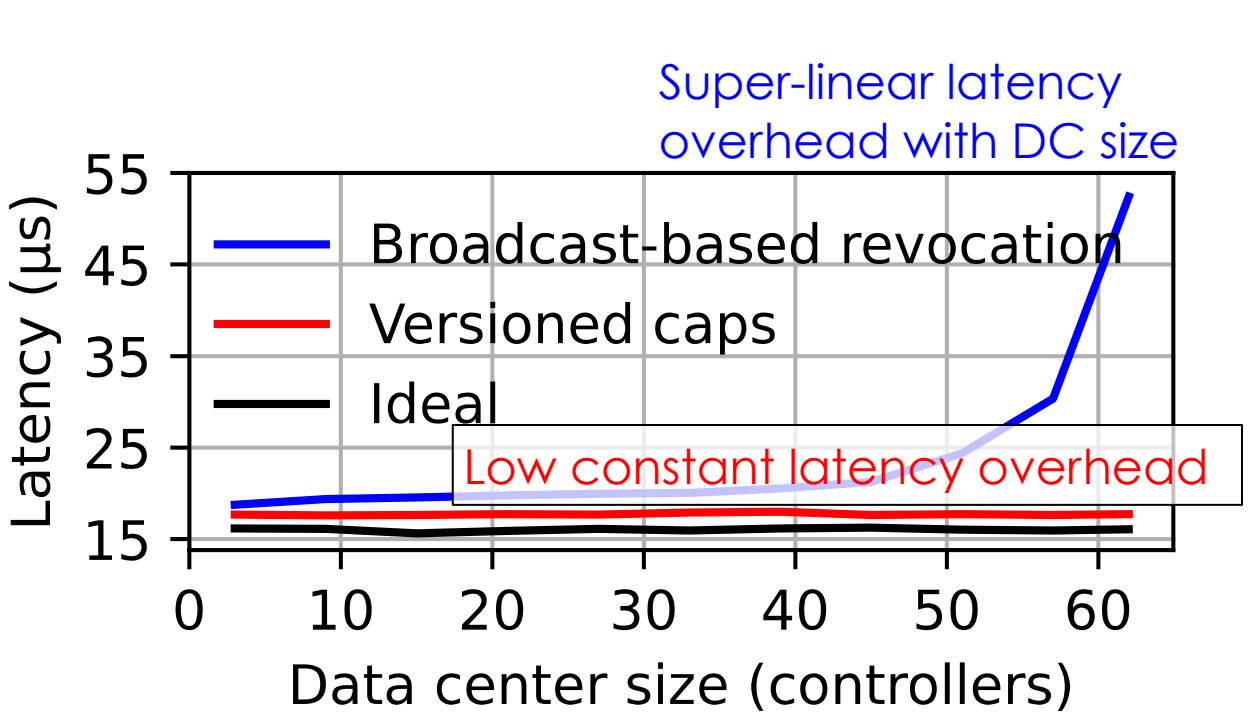

Network traffic [TB/s]
1000
500
0
Broadcast-based
revocation (20% ctrls)
0
25k
50k
75k
100k
Data center size (controllers)

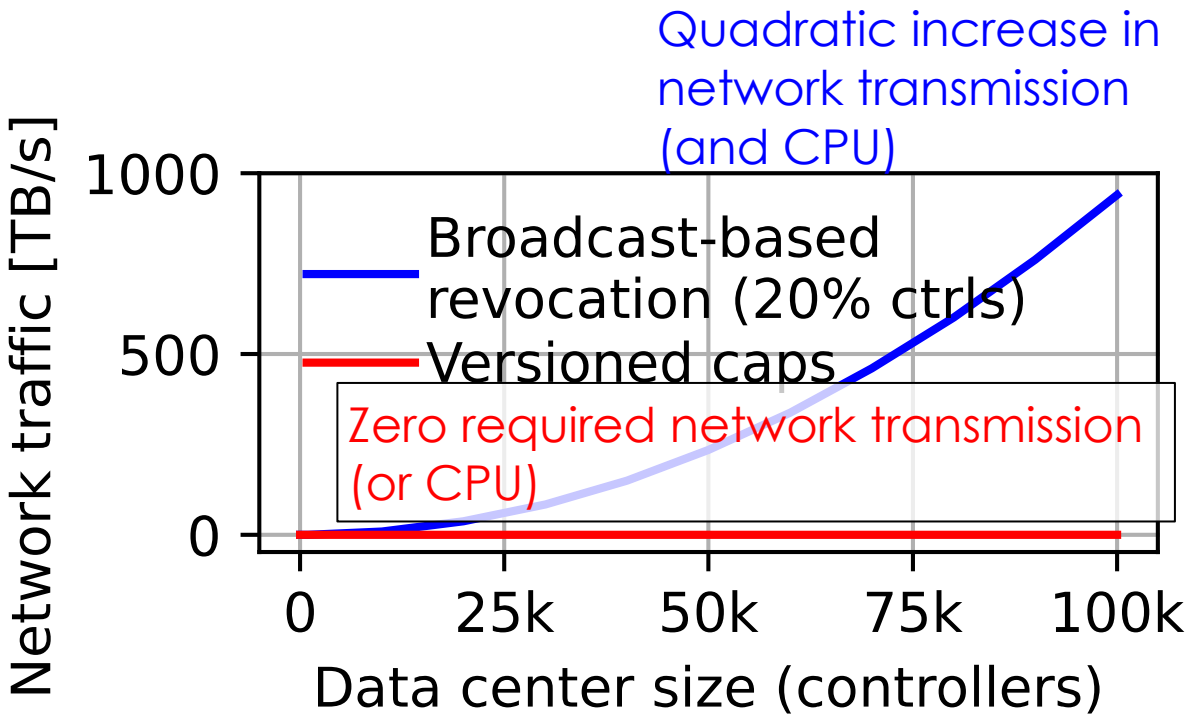
Evaluation


Super-linear latency overhead with DC size
Latency (µs)
55
45
35
25
15
Broadcast-based revocation
Versioned caps
Ideal
Low constant latency overhead
0
10
20
30
40
50
60
Data center size (controllers)

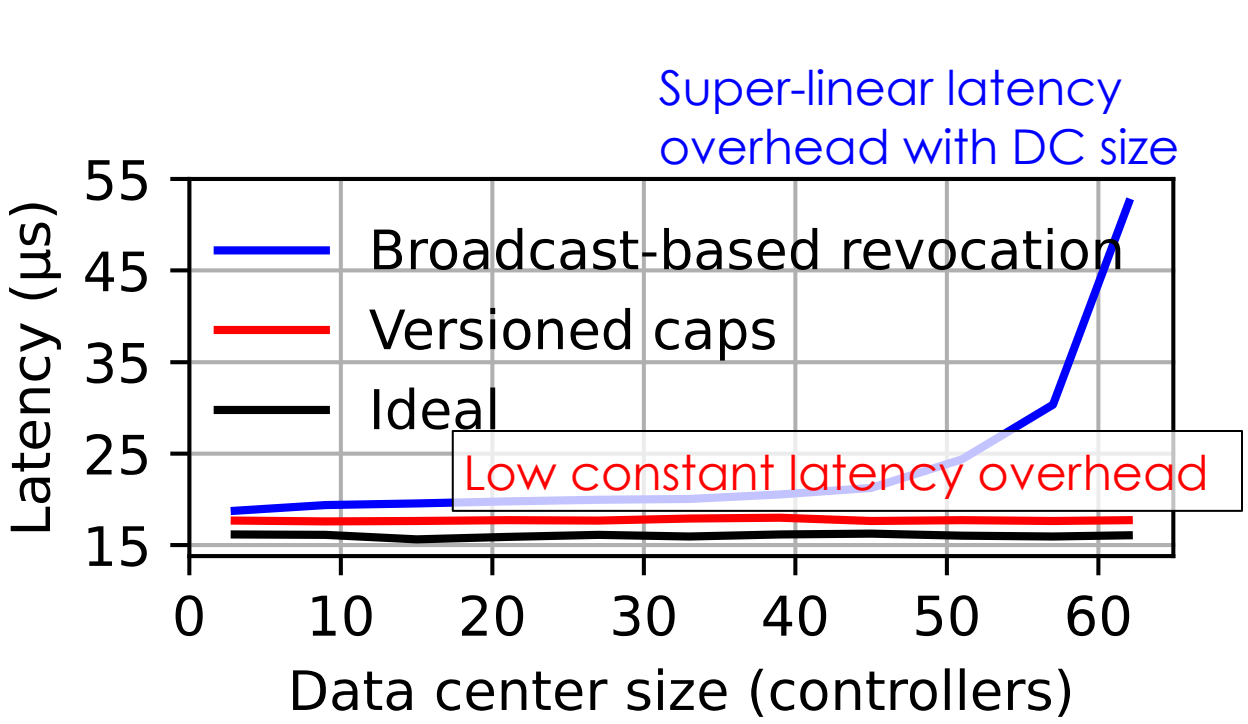

Quadratic increase in network transmission (and CPU)
Network traffic [TB/s]
1000
500
0
Broadcast-based revocation (20% ctrls)
0
25k
50k
75k
100k
Data center size (controllers)

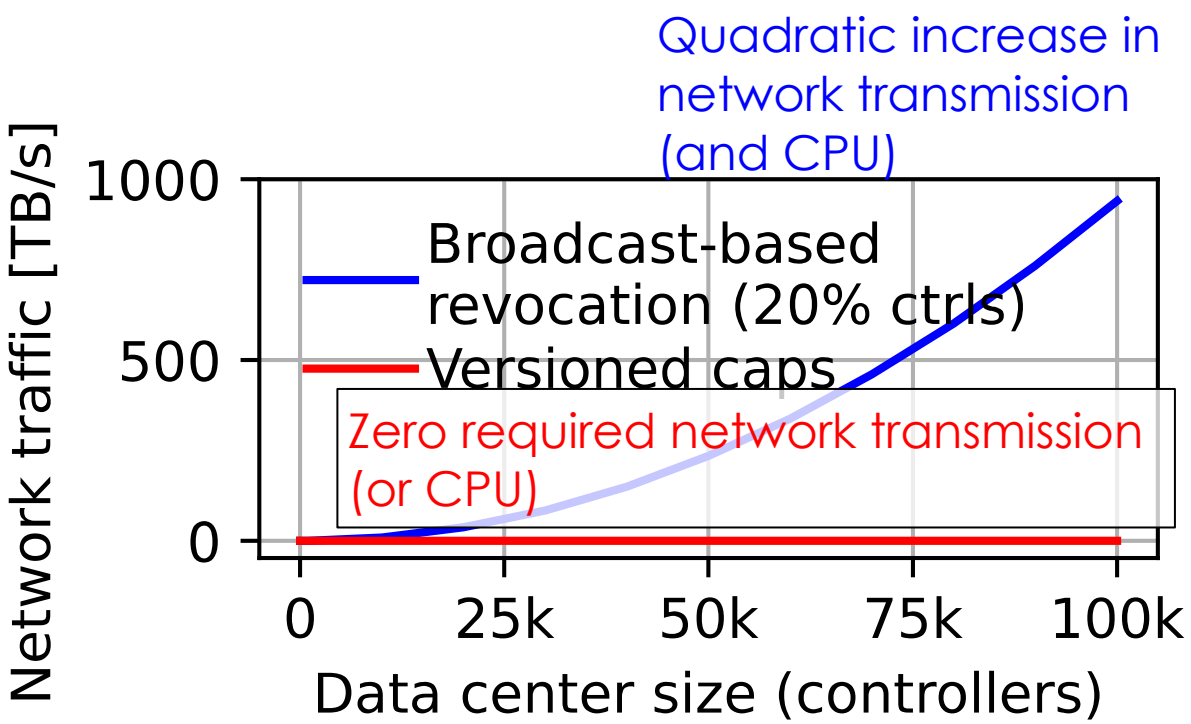
Evaluation


Super-linear latency overhead with DC size
Latency (µs)
55
45
35
25
15
0
10
20
30
40
50
60
Data center size (controllers)
Broadcast-based revocation
Versioned caps
Ideal
Low constant latency overhead


Quadratic increase in network transmission (and CPU)
Network traffic [TB/s]
1000
500
0
0
25k
50k
75k
100k
Data center size (controllers)
Broadcast-based revocation (20% ctrls)
Versioned caps
Zero required network transmission (or CPU)

Evaluation





- Constant-cost, low µs-scale cap operations (eval in paper)

Conclusion

Conclusion

- Three key innovations
  - Owner-based metadata sharding, versioned capabilities and guards (detail in paper)

Conclusion

- Three key innovations
  - Owner-based metadata sharding, versioned capabilities and guards (detail in paper)
- To achieve all three cloud-scale challenges

Conclusion

- Three key innovations
  - Owner-based metadata sharding, versioned capabilities and guards (detail in paper)
- To achieve all three cloud-scale challenges
  - Failure isolation

Conclusion

- Three key innovations
  - Owner-based metadata sharding, versioned capabilities and guards (detail in paper)
- To achieve all three cloud-scale challenges
  - Failure isolation
  - Constant-cost, µs-scale cap operations (eval in paper)

Conclusion

- Three key innovations
  - Owner-based metadata sharding, versioned capabilities and guards (detail in paper)
- To achieve all three cloud-scale challenges
  - Failure isolation
  - Constant-cost, µs-scale cap operations (eval in paper)
  - Constant system overheads (network transmission, CPU usage, latency overheads) regardless of DC size

Conclusion

- Three key innovations
  - Owner-based metadata sharding, versioned capabilities and guards (detail in paper)
- To achieve all three cloud-scale challenges
  - Failure isolation
  - Constant-cost, µs-scale cap operations (eval in paper)
  - Constant system overheads (network transmission, CPU usage, latency overheads) regardless of DC size

Otto White 😄 👍
otto.white20@imperial.ac.uk