Storage Services

Yves Goeleven

Solution Architect - Particular Software

- Shipping software since 2001
- Azure MVP since 2010
- Co-founder & board member AZUG
- NServiceBus & MessageHandler

Used azure storage?

Agenda

Deep Dive Architecture

- Global Namespace
- Front End Layer
- Partition Layer
- Stream Layer
- Fault Tolerance

This talk is based on an academic whitepaper

Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency

Brad Calder, Ju Wang, Aaron Ogus, Niranjan Nilakantan, Arild Skjolsvold, Sam McKelvie, Yikang Xu, Shashwat Srivastav, Jiesheng Wu, Huseyin Simitci, Jaidev Haridas, Chakravarthy Uddaraju, Hemal Khatri, Andrew Edwards, Vaman Bedekar, Shane Mainali, Rafay Abbasi, Arpit Agarwal, Mian Fahim ul Haq, Muhammad Ikram ul Haq, Deepali Bhardwaj, Sowmya Dayanand, Anitha Adusumilli, Marvin McNett, Sriram Sankaran, Kavitha Manivannan, Leonidas Rigas

Microsoft

http://sigops.org/sosp/sosp11/current/2011-Cascais/printable/11-calder.pdf

Let’s dive in

Global namespace

Global namespace

Every request gets sent to

- http(s)://**AccountName.Service**.core.windows.net/**PartitionName**/**ObjectName**

Global namespace

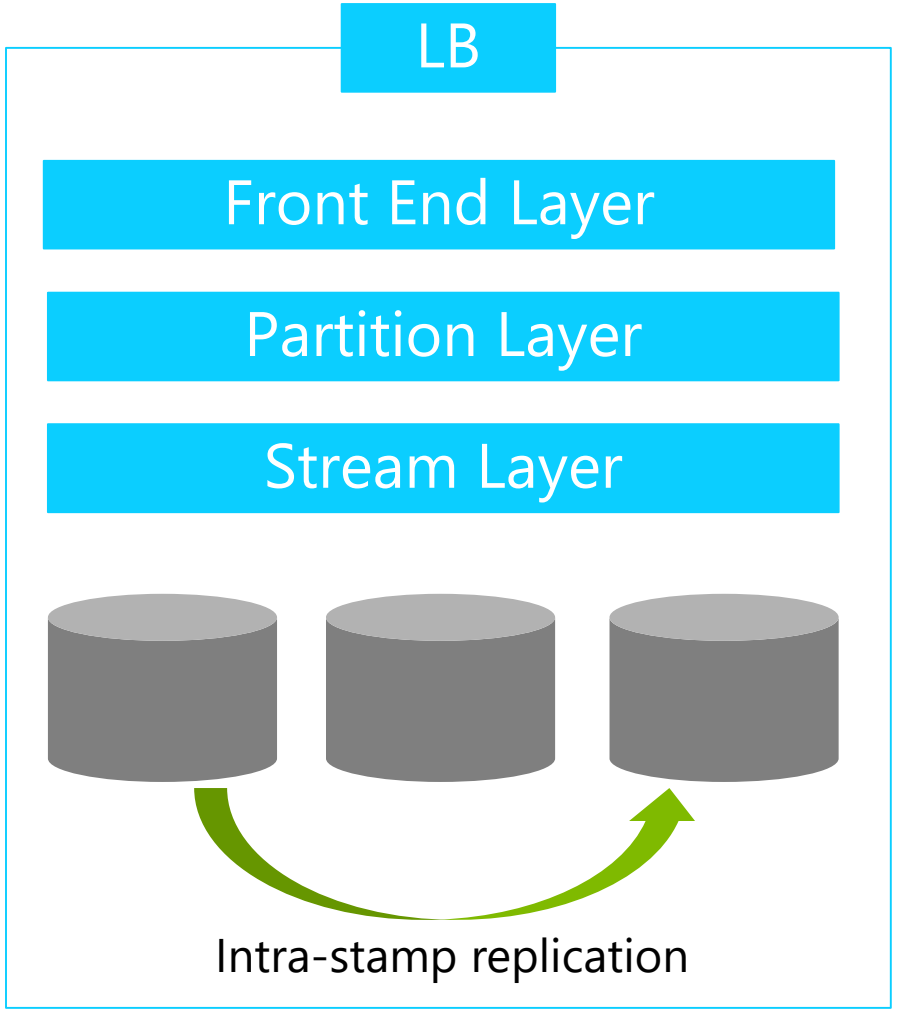
DNS based service location

- http(s)://**AccountName.Service**.core.windows.net/PartitionName/ObjectName
- DNS resolution
- Datacenter (f.e. Western Europe)
- Virtual IP of a Stamp

Stamp

Scale unit

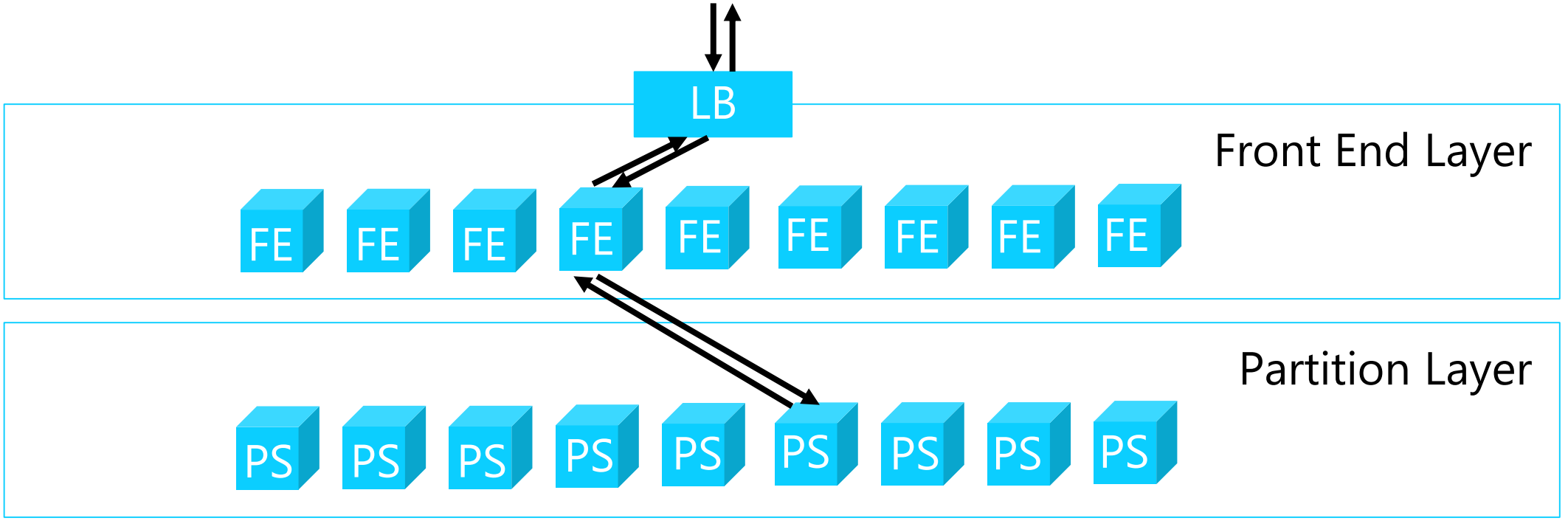
- A windows azure deployment
- 3 layers
  - Typically 20 racks each
  - Roughly 360 XL instances in total
- Instances with disks attached
  - 2 Petabyte (before june 2012)
  - 30 Petabyte



Incoming read request

Arrives at Front End

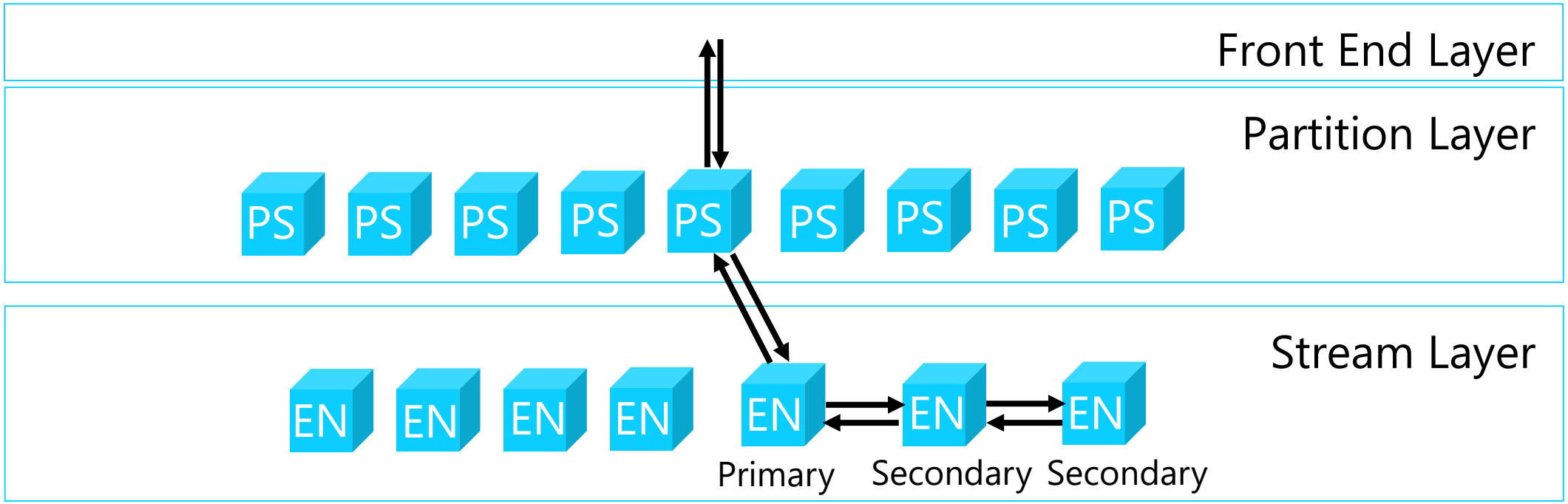
- Routed to partition layer
- http(s)://**AccountName**.Service.core.windows.net/**PartitionName**/**ObjectName**
- Reads usually from memory



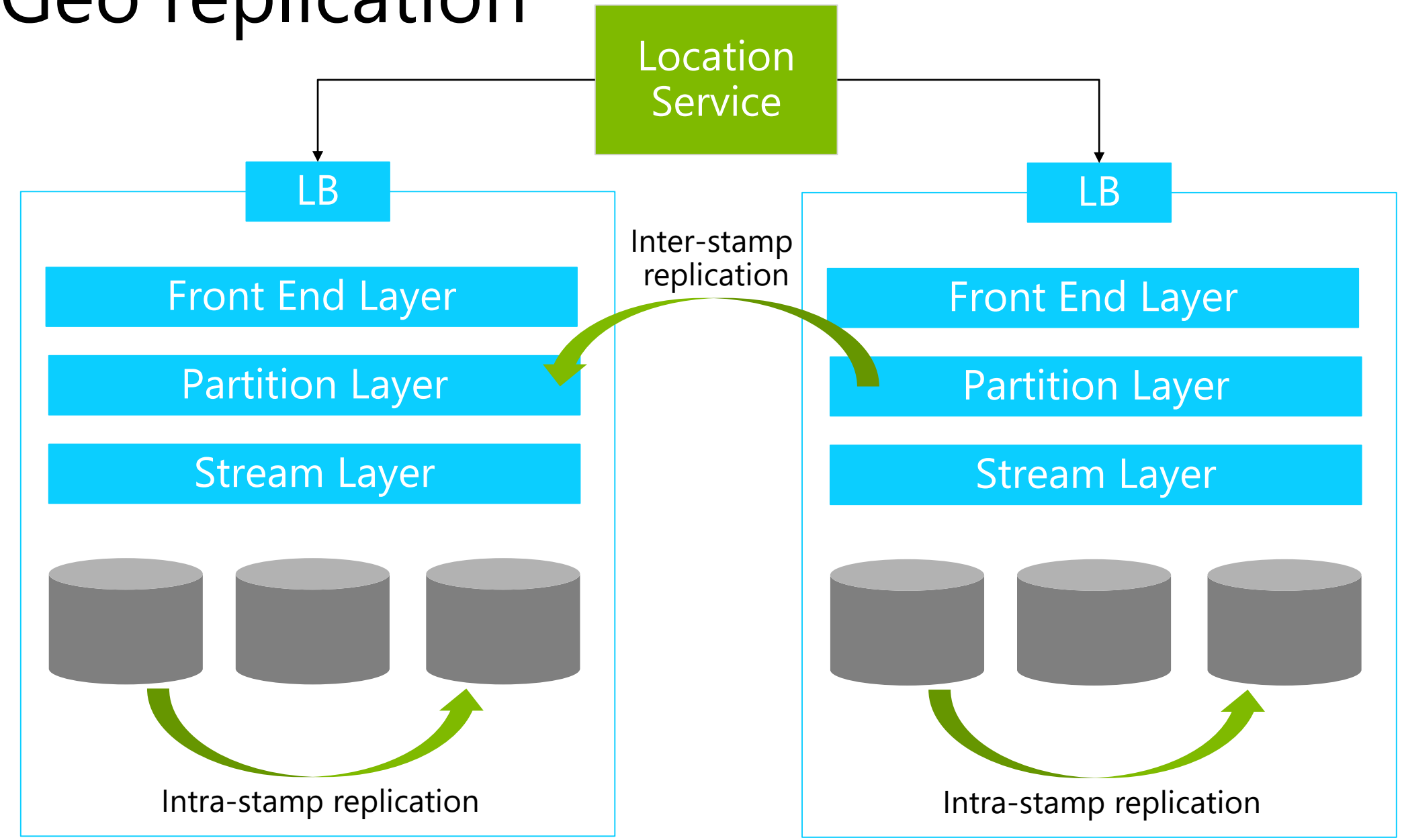
Incoming Write request

Written to stream layer

- Synchronous replication
- 1 Primary Extend Node
- 2 Secondary Extend Nodes



Geo replication



Location Service

Stamp Allocation

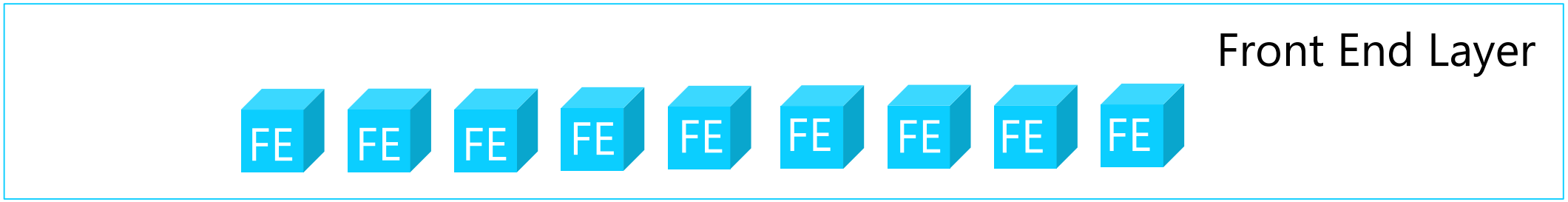
- Manages DNS records
- Determines active/passive stamp configuration
  - Assigns accounts to stamps (ARM)
  - Geo Replication
  - Disaster Recovery
  - Load Balancing
  - Account Migration

Front End Layer

Front End Nodes

Stateless role instances

- Authentication & Authorization
  - Shared Key (storage key)
  - Shared Access Signatures
- Routing to Partition Servers
  - Based on PartitionMap (in memory cache)
- Versioning
  - x-ms-version: 2011-08-18
- Throttling



Partition Map

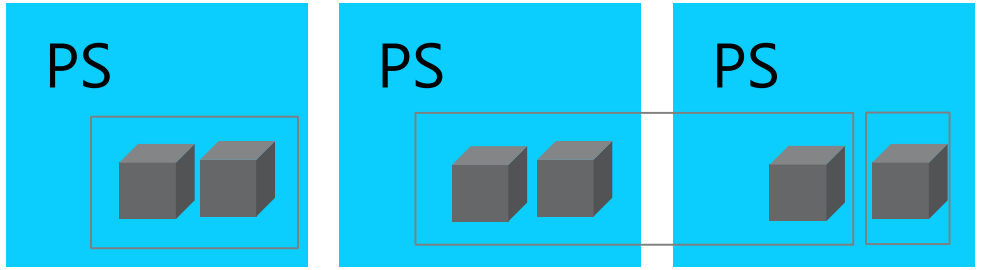
Determine partition server

- Used By Front End Nodes
- Managed by Partition Manager (Partition Layer)
- Ordered Dictionary
  - Key: http(s)://**AccountName**.**Service**.core.windows.net/**PartitionName**/**ObjectName**
  - Value: Partition Server
- Range Partition
  - Split across partition servers

Practical examples

Blob

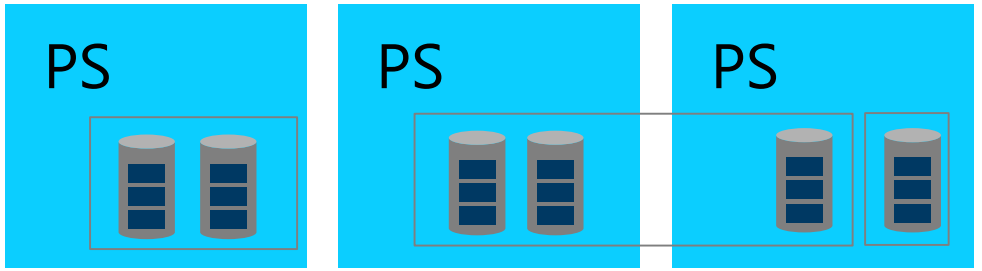
- Each blob is a partition
- Blob uri determines range
- Range only matters for listing
- Same container: More likely same range
  - https://**myaccount**.blobs.core.windows.net**/mycontainer/subdir/file1.jpg**
  - https://**myaccount**.blobs.core.windows.net**/mycontainer/subdir/file2.jpg**
- Other container: Less likely
  - https://**myaccount**.blobs.core.windows.net**/othercontainer/subdir/file1.jpg**



Practical examples

Queue

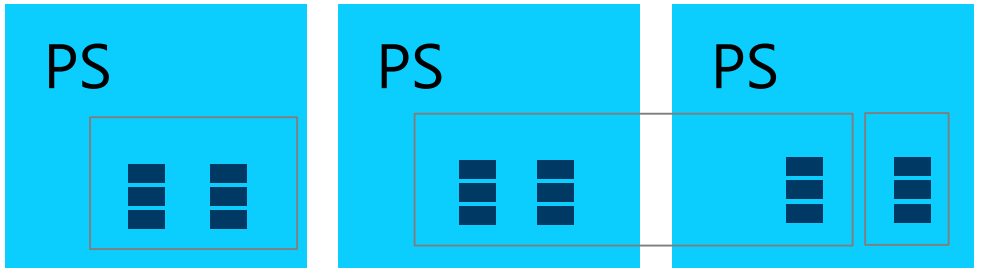
- Each queue is a partition
- Queue name determines range
- All messages same partition
- Range only matters for listing queues



Practical examples

Table

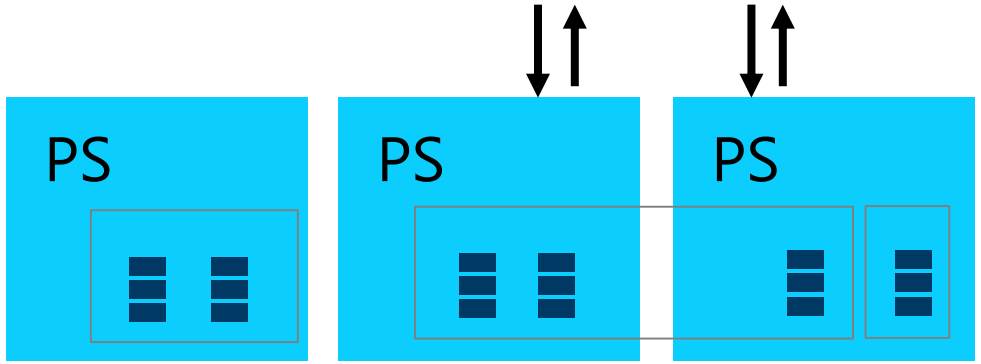
- Each group of rows with same PartitionKey is a partition
- Ordering within set by RowKey
- Range matters a lot!
- Expect continuation tokens!



Continuation tokens

Requests across partitions

- All on same partition server
  - 1 result
- Across multiple ranges (or to many results)
  - Sequential queries required
  - 1 result + continuation token
  - Repeat query with token until done
- Will almost never occur in development!
  - As dataset is typically small



Partition Layer

Partition layer

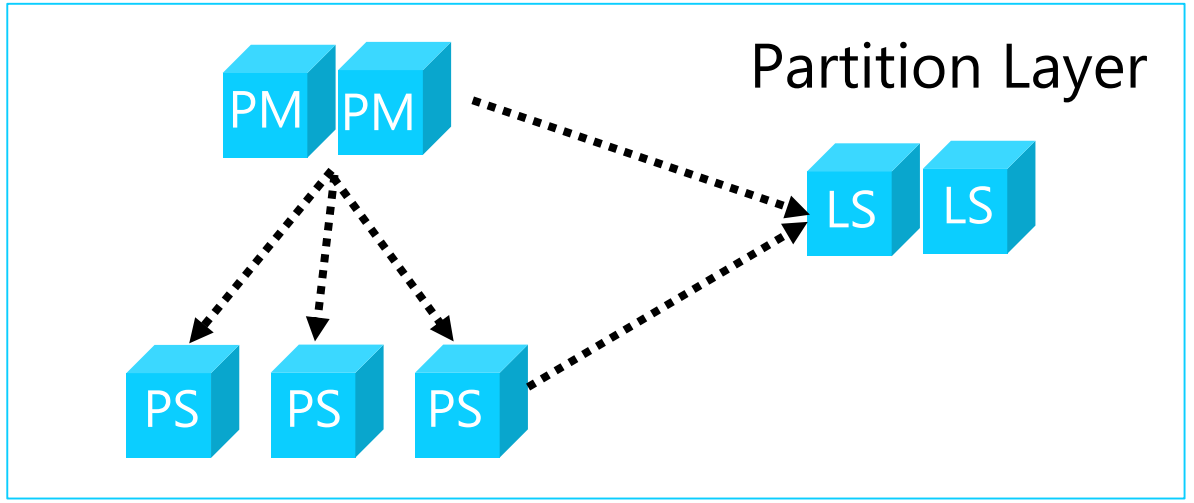
Dynamic Scalable Object Index

- Object Tables
  - System defined data structures
  - Account, Entity, Blob, Message, Schema, PartitionMap
- Dynamic Load Balancing
  - Monitor load to each part of the index to determine hot spots
  - Index is dynamically split into thousands of Index Range Partitions based on load
  - Index Range Partitions are automatically load balanced across servers to quickly adapt to changes in load
  - Updates every 15 seconds

Partition layer

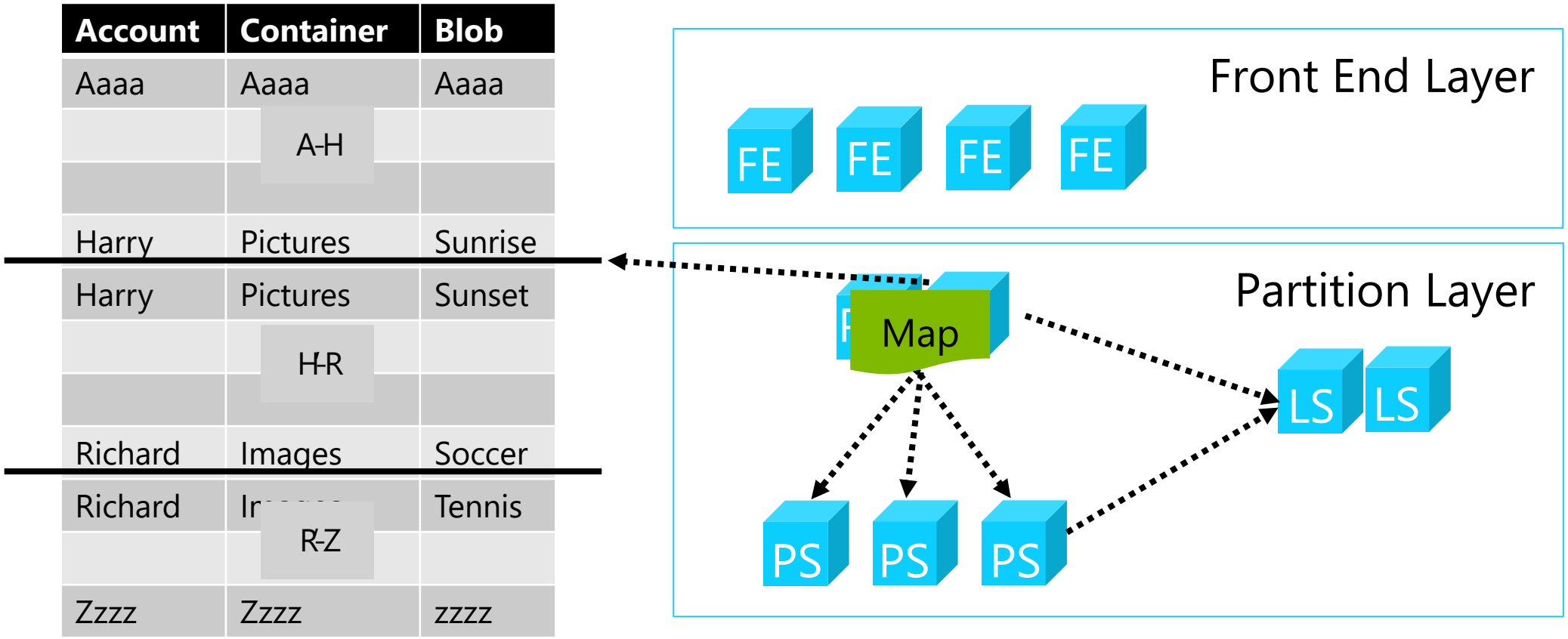
Consists of

- Partition Master
  - Distributes Object Tables
  - Partitions across Partition Servers
  - Dynamic Load Balancing
- Partition Server
  - Serves data for exactly 1 Range Partition
  - Manages Streams
- Lock Server
  - Master Election
  - Partition Lock



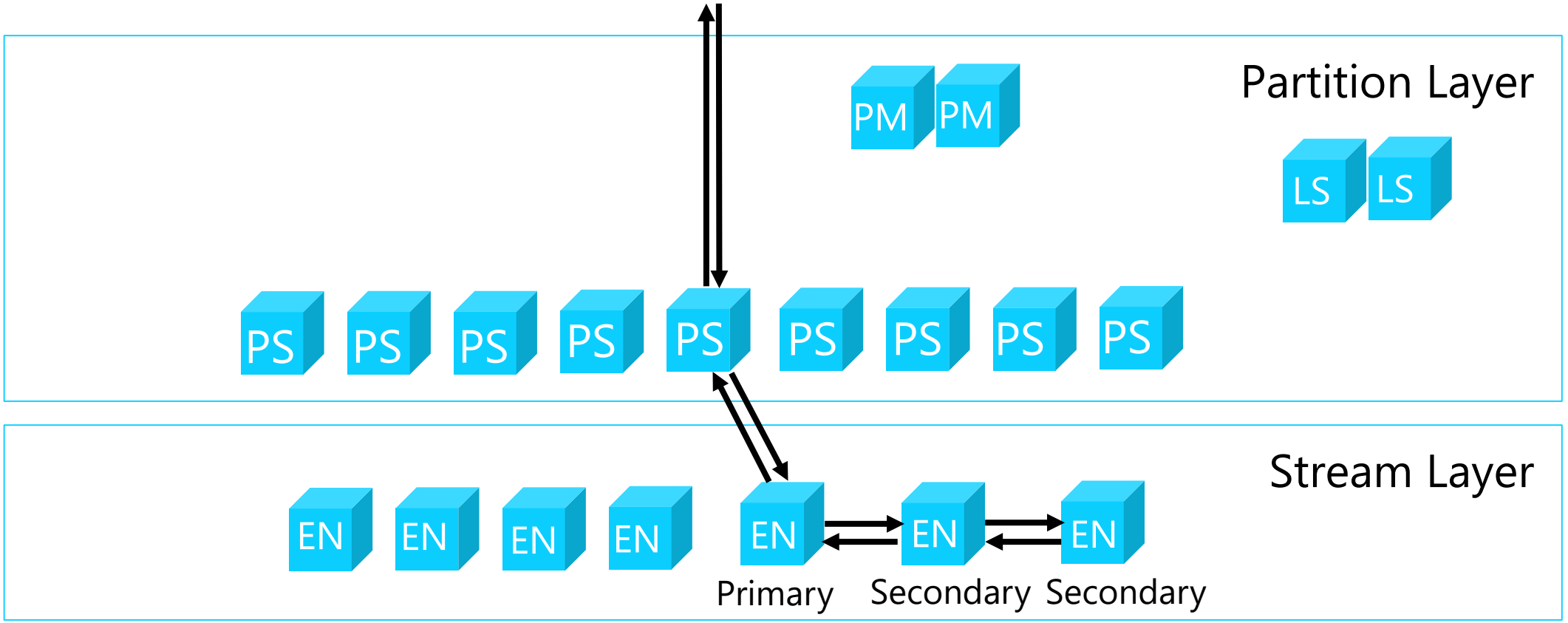
Dynamic Scalable Object Index

How it works



Dynamic Load Balancing

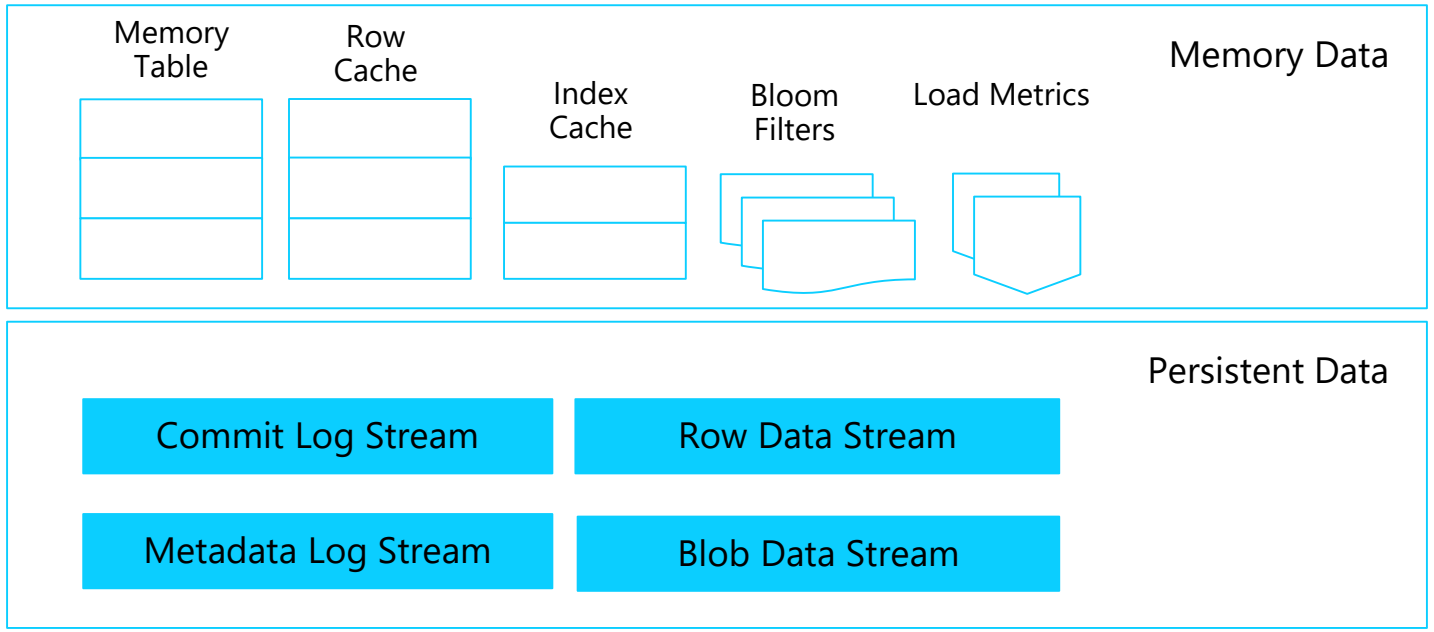
Not in the critical path!



Partition Server

Architecture

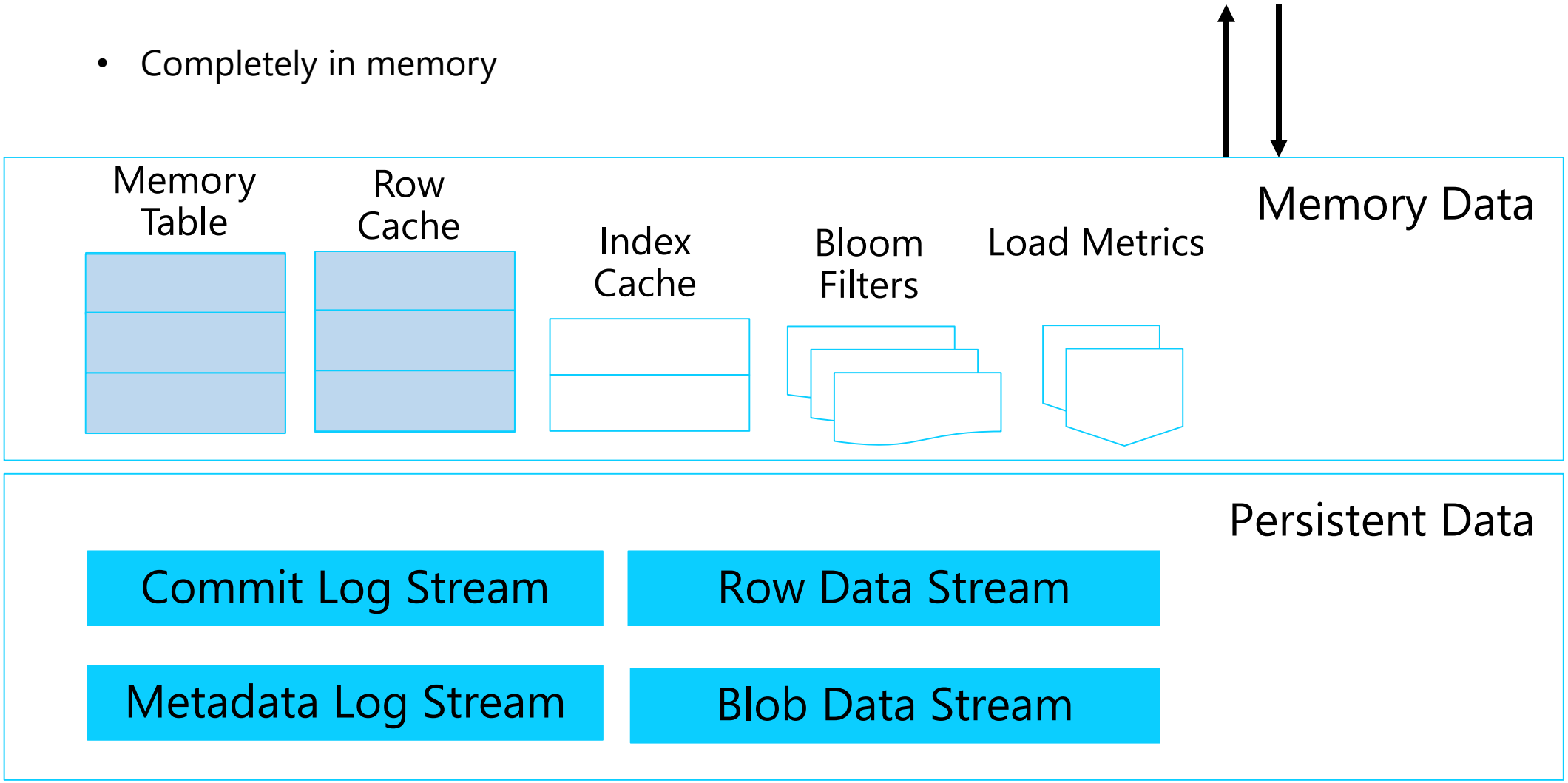
- Log Structured Merge-Tree
  - Easy replication
- Append only persisted streams
  - Commit log
  - Metadata log
  - Row Data (Snapshots)
  - Blob Data (Actual blob data)
- Caches in memory
  - Memory table (= Commit log)
  - Row Data
  - Index (snapshot location)
  - Bloom filter
  - Load Metrics



Incoming Read request

HOT Data

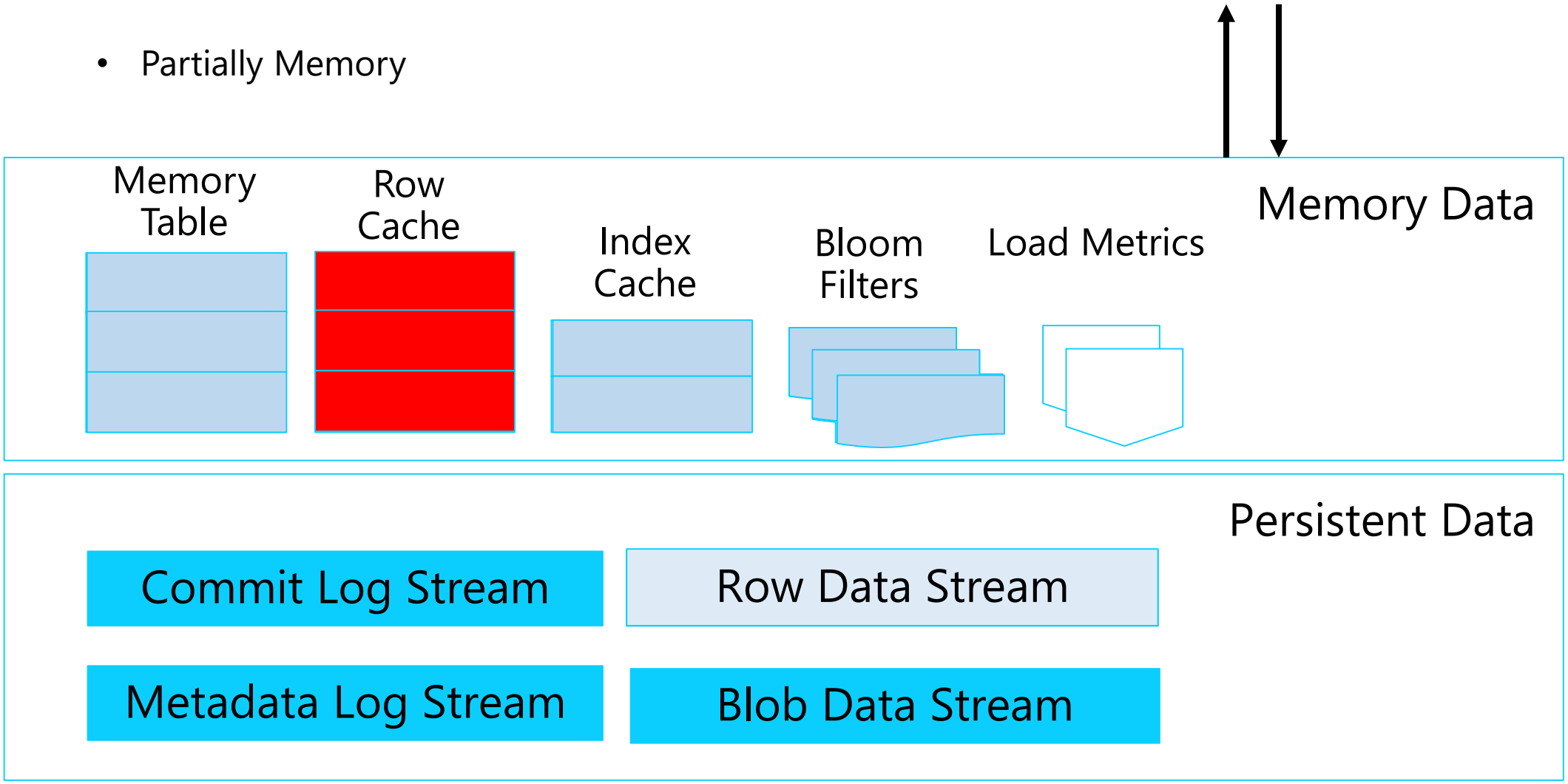
- Completely in memory



Incoming Read request

COLD Data

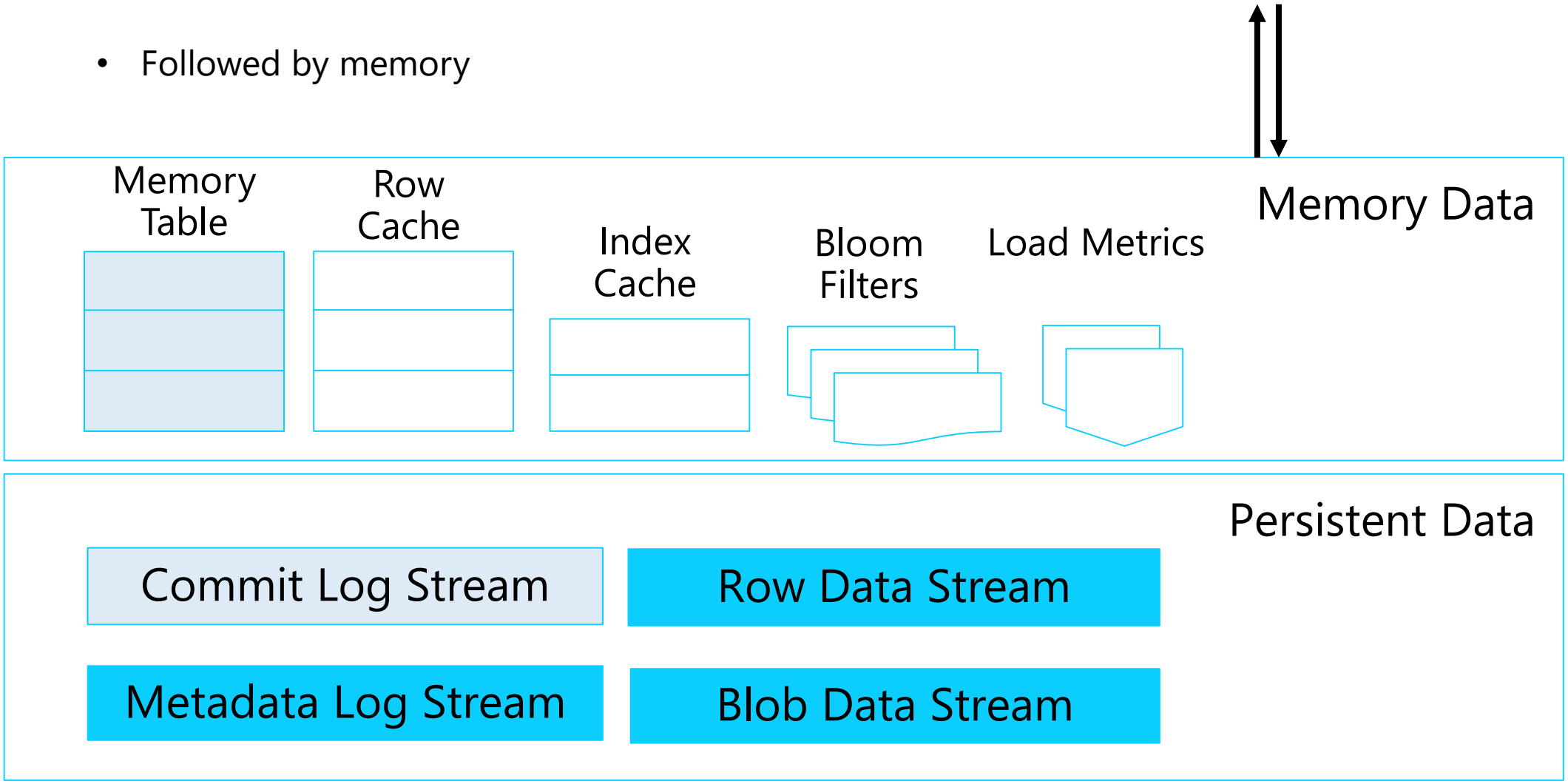
- Partially Memory



Incoming Write Request

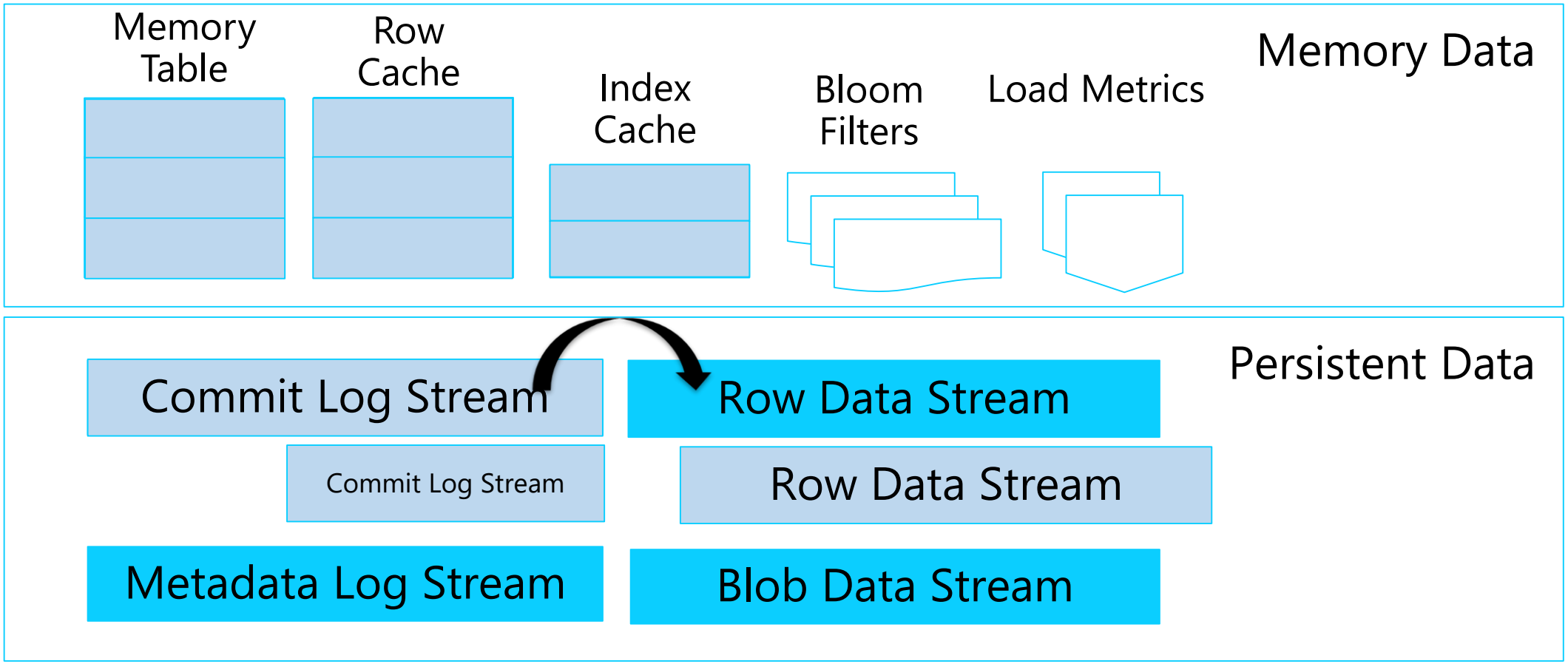
Persistent First

- Followed by memory



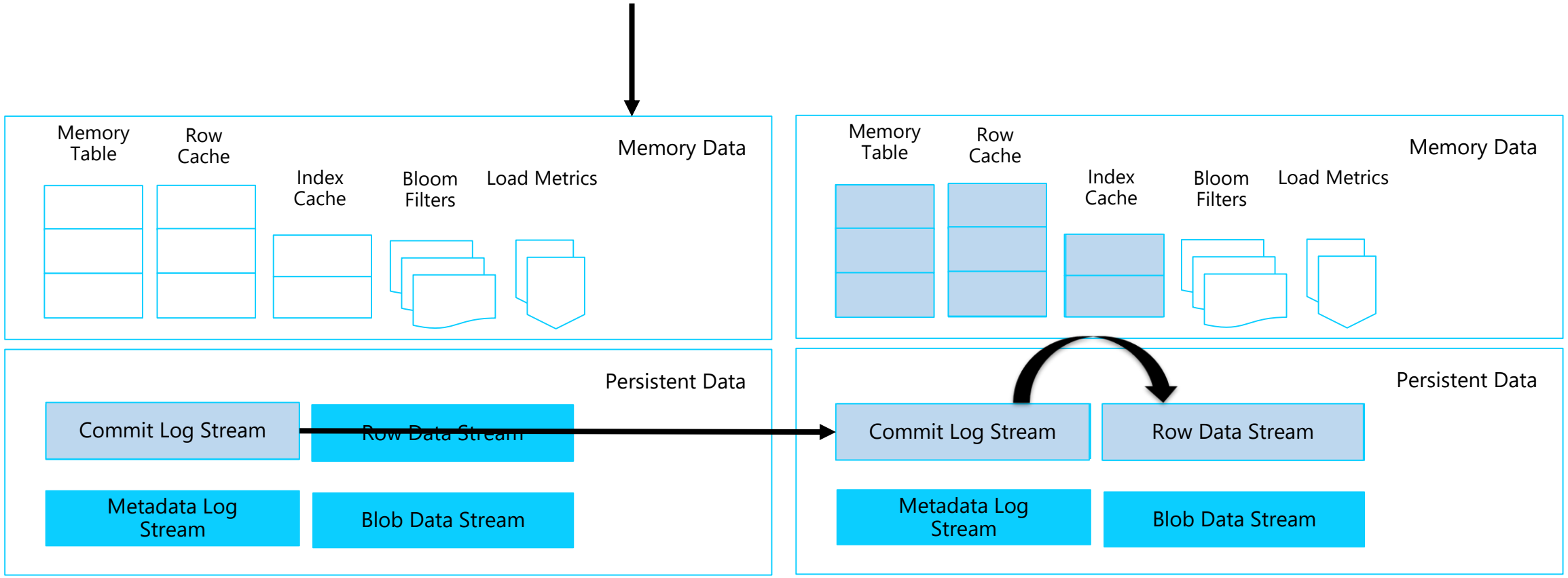
Checkpoint process

Not in critical path!



Geo replication process

Not in critical path!



Stream Layer

Stream layer

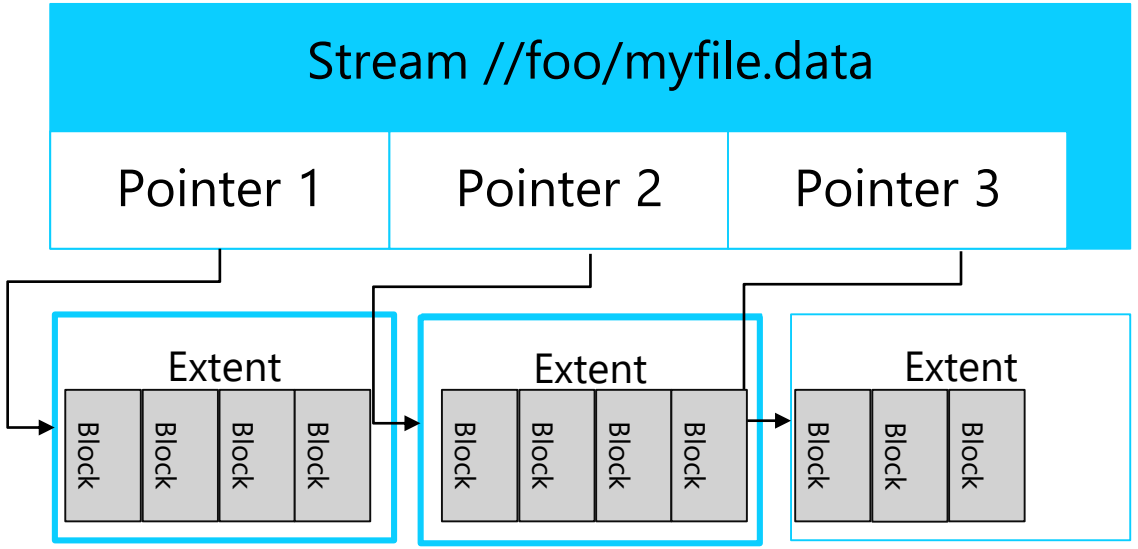
Append-Only Distributed File System

- Streams are very large files
  - File system like directory namespace
- Stream Operations
  - Open, Close, Delete Streams
  - Rename Streams
  - Concatenate Streams together
  - Append for writing
  - Random reads

Stream layer

Concepts

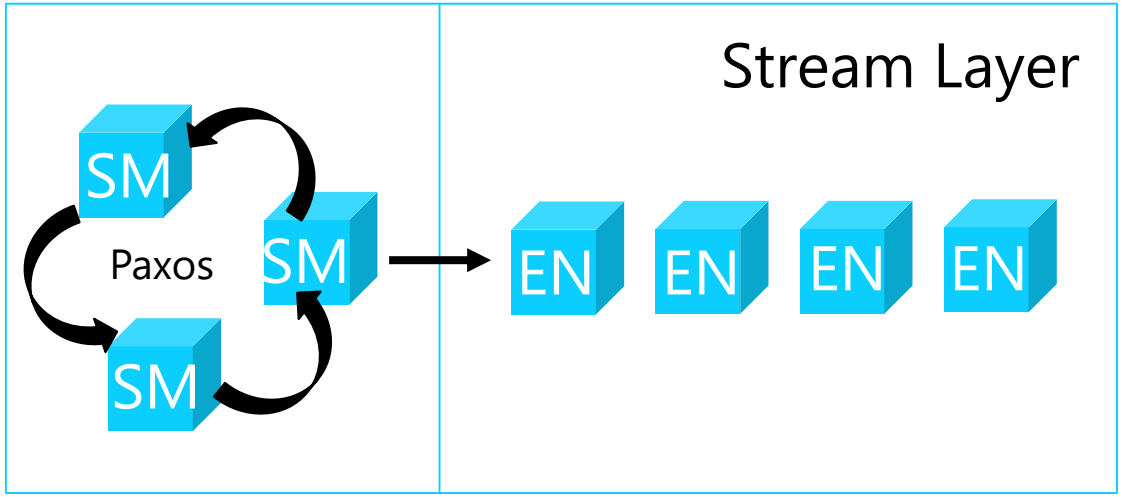
- Block
  - Min unit of write/read
  - Checksum
  - Up to N bytes (e.g. 4MB)
- Extent
  - Unit of replication
  - Sequence of blocks
  - Size limit (e.g. 1GB)
  - Sealed/unsealed
- Stream
  - Hierarchical namespace
  - Ordered list of pointers to extents
  - Append/Concatenate



Stream layer

Consists of

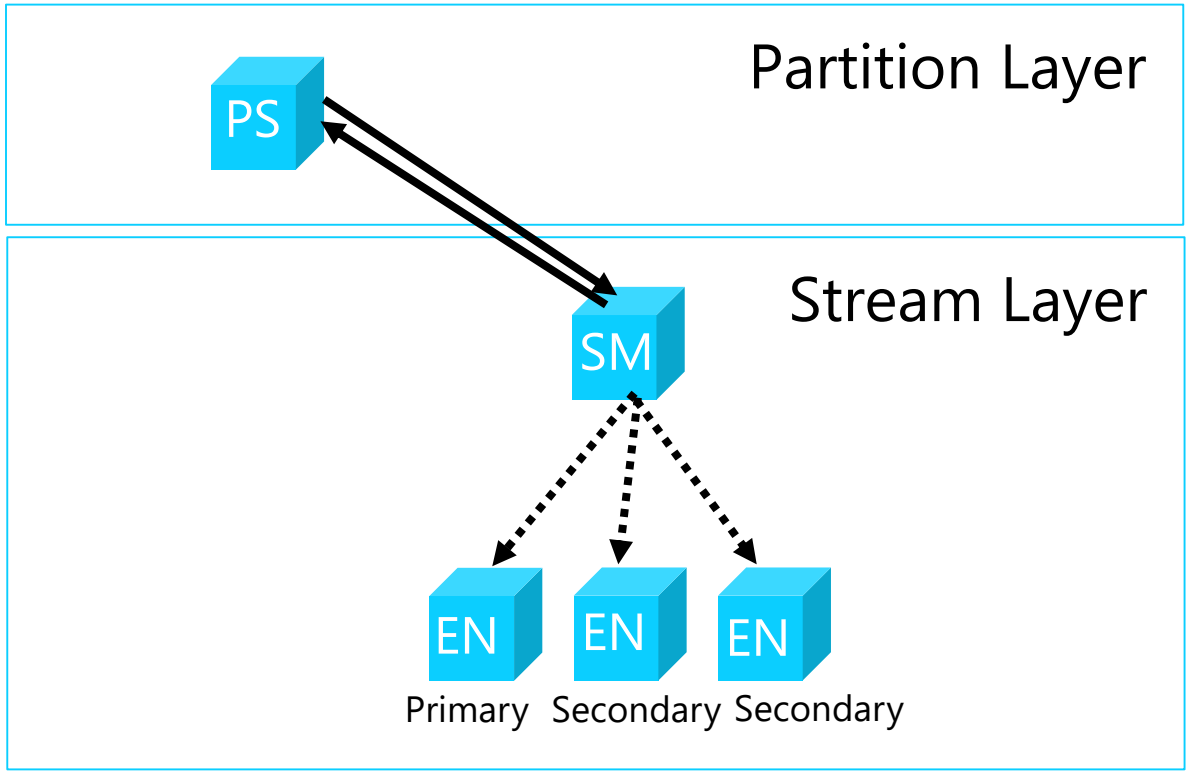
- Stream Master
  - Paxos System
  - Distributed consensus protocol
- Extend Nodes
  - Attached Disks
  - Manages Extents
  - Optimization Routines



Extent creation

Allocation process

- Partition Server
  - Request extent / stream creation
- Stream Master
  - Chooses 3 random extend nodes based on available capacity
  - Chooses Primary & Secondary's
  - Allocates replica set
- Random allocation
  - To reduce Mean Time To Recovery



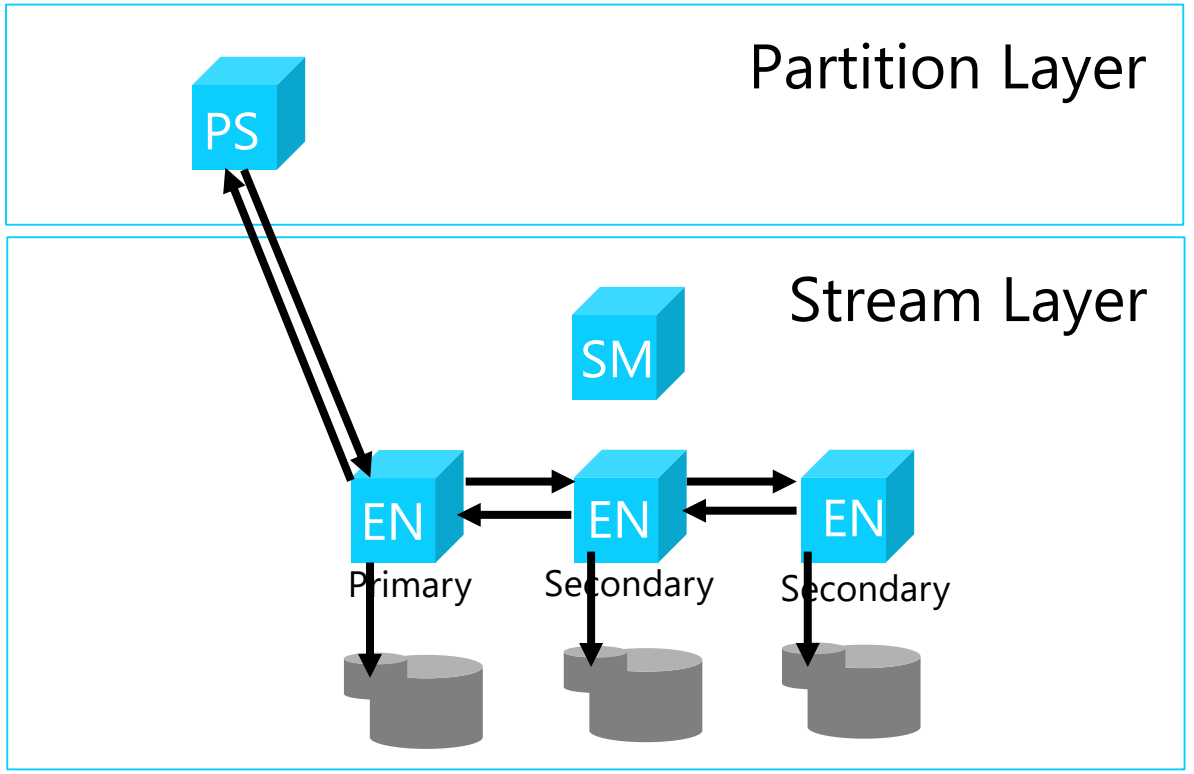
Replication Process

Bitwise equality on all nodes

- Synchronous
- Append request to primary
- Written to disk, checksum
- Offset and data forwarded to secondaries
- Ack

Journaling

- Dedicated disk (spindle/ssd) for writes
- Simultaneously copied to data disk
  (from memory or journal)



Fault Tolerance

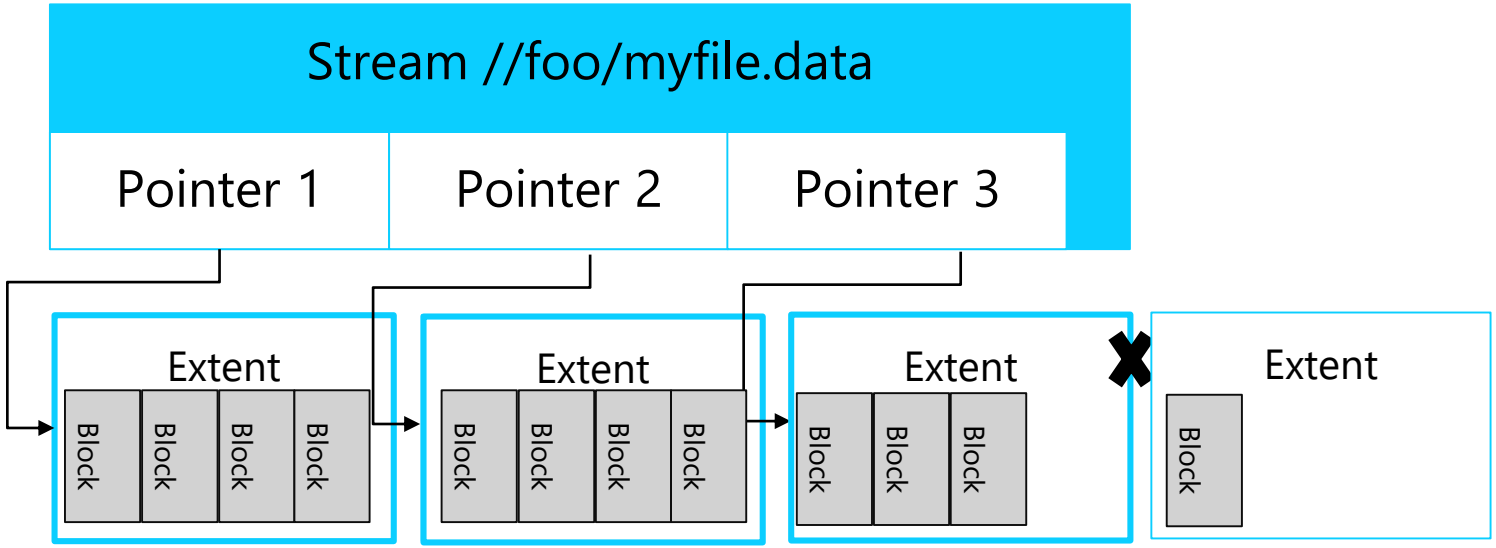
Dealing with failure

Ack from primary lost going back to partition layer

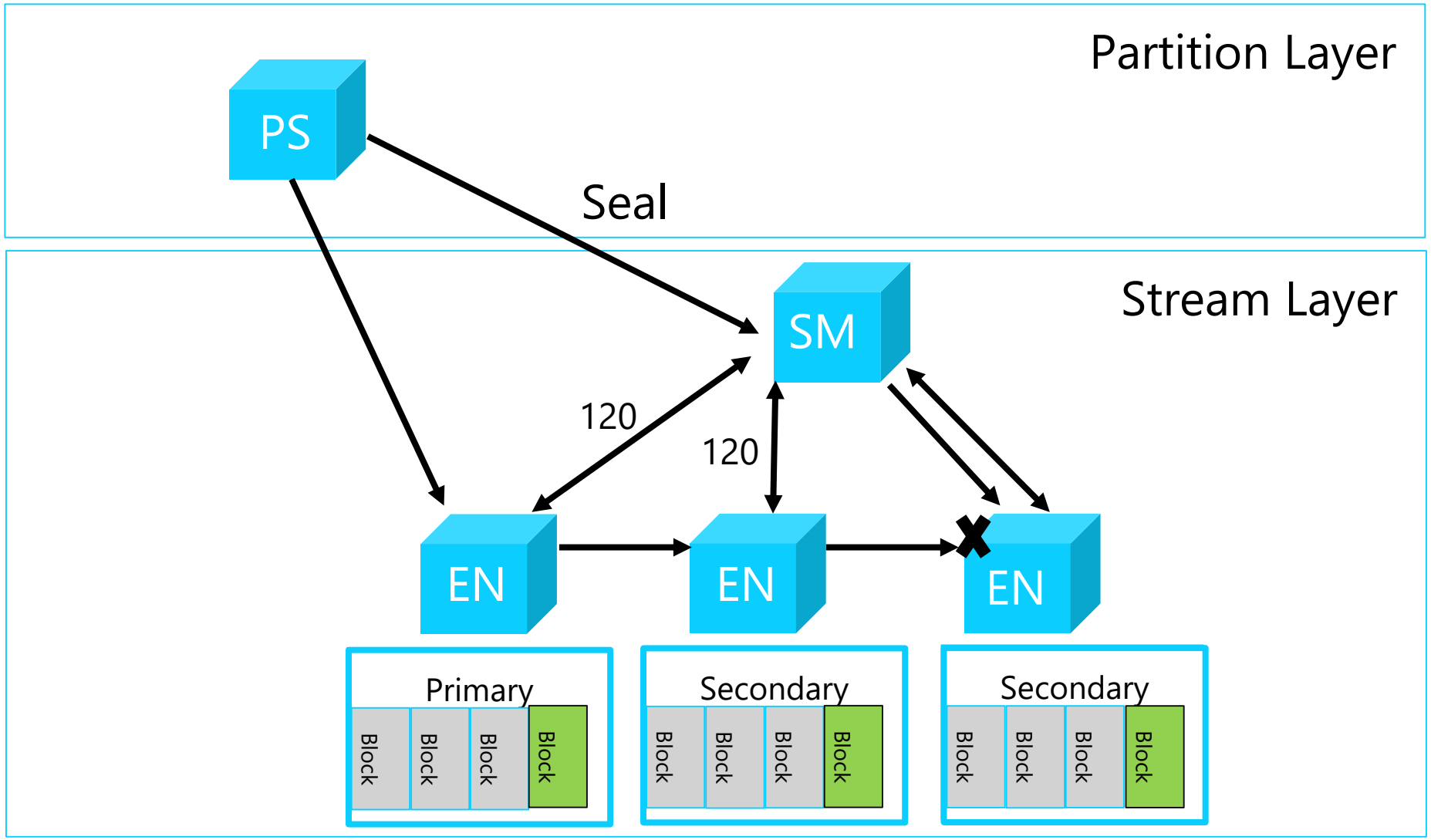
- Retry from partition layer
- can cause multiple blocks to be appended (duplicate records)

Unresponsive/Unreachable Extent Node (EN)

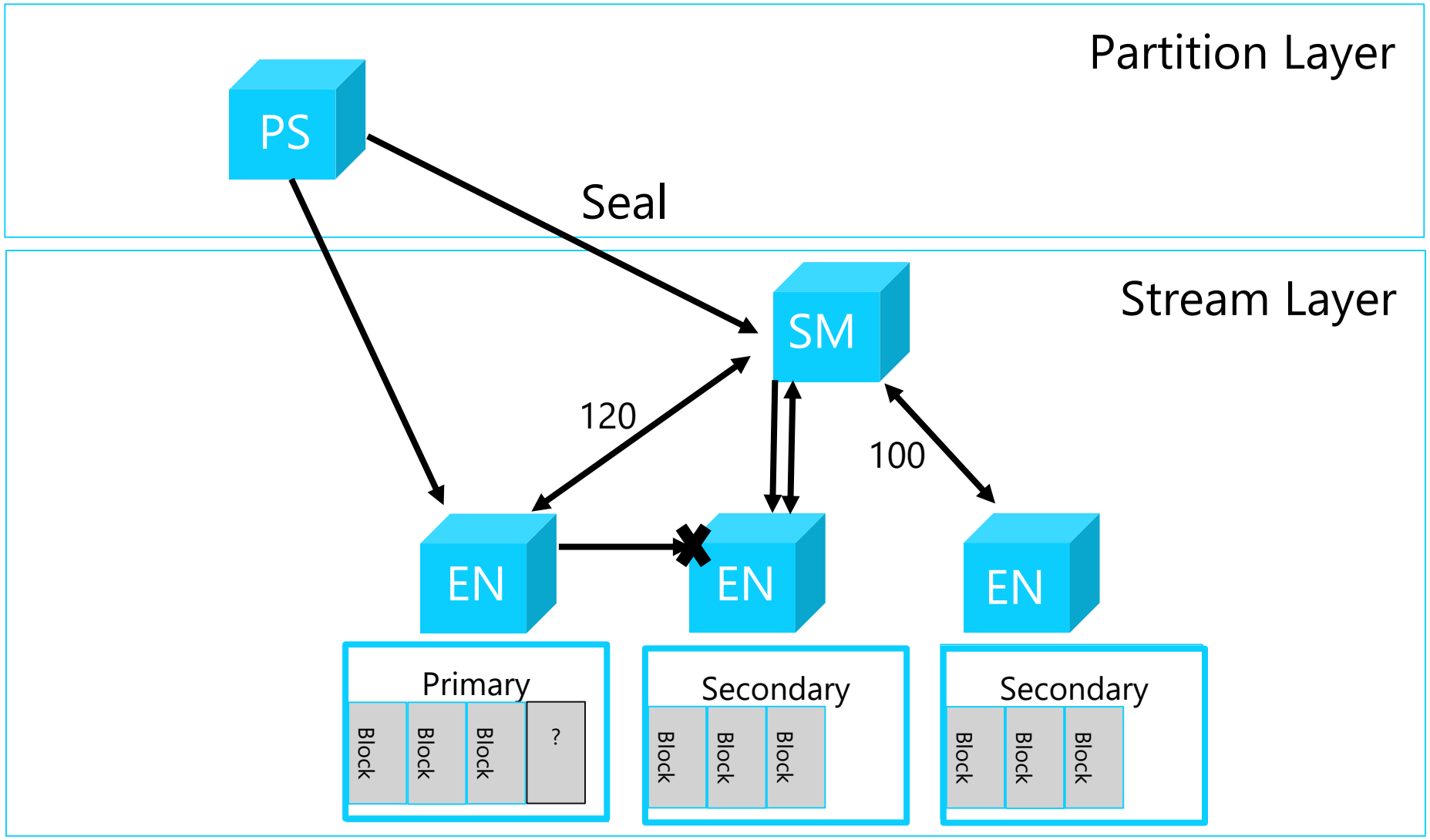
- Seal the failed extent & allocate a new extent and append immediately



Replication Failures (Scenario 1)



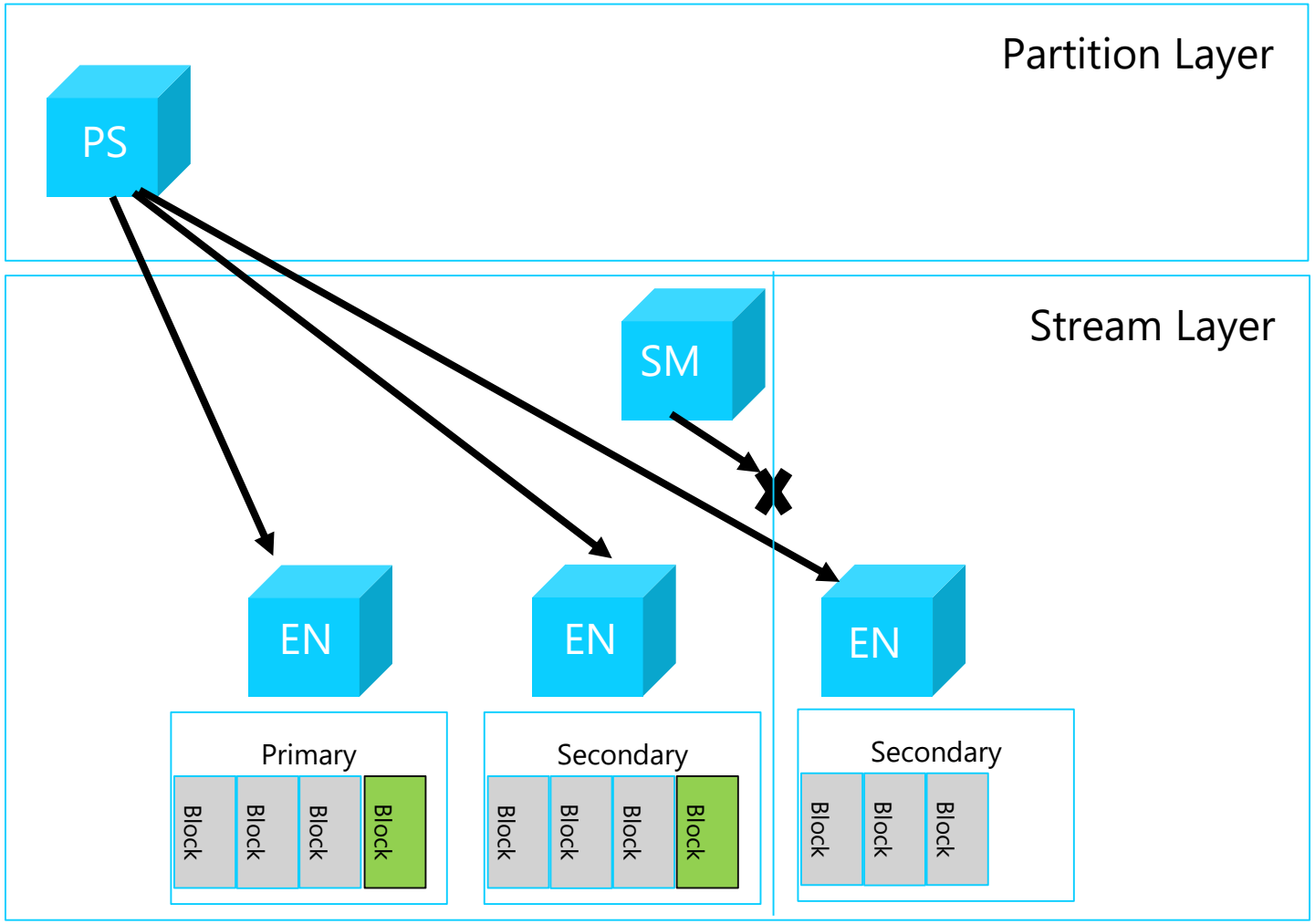
Replication Failures (Scenario 2)



Network partitioning during read

Data Stream

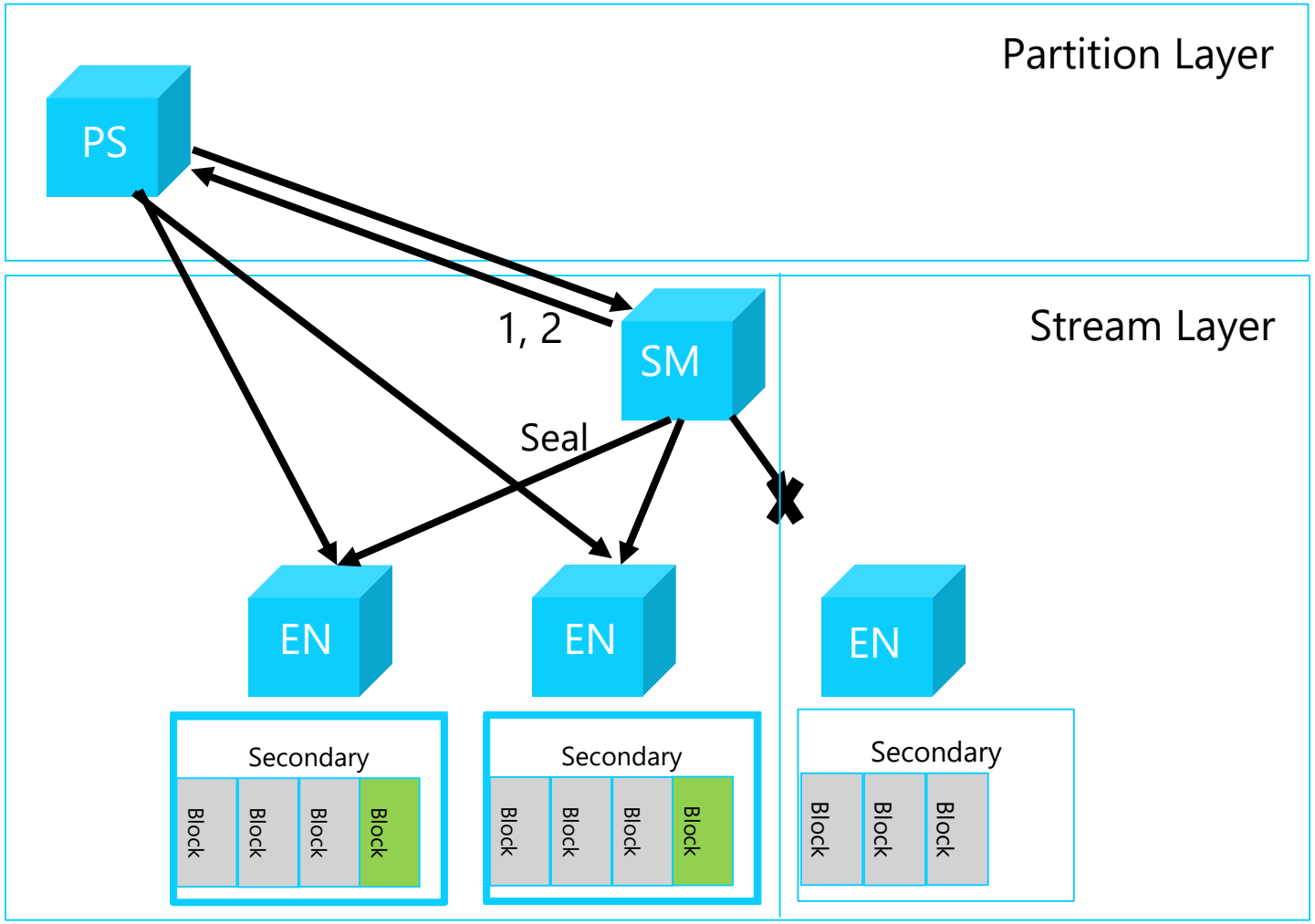
- Partition Layer only reads from offsets returned from successful appends
- Committed on all replicas
- Row and Blob Data Streams
- Offset valid on any replica
- Safe to read



Network partitioning during read

Log Stream

- Logs are used on partition load
- Check commit length first
- Only read from
  - Unsealed replica if all replicas have the same commit length
  - A sealed replica



Garbage Collection

Downside to this architecture

- Lot's of wasted disk space
  - 'Forgotten' streams
  - 'Forgotten' extends
  - Unused blocks in extends after write errors
- Garbage collection required
  - Partition servers mark used streams
  - Stream manager cleans up unreferenced streams
  - Stream manager cleans up unreferenced extends
- Stream manager performs erasure coding
  - Reed Solomon algorithm
  - Allows recreation of cold streams after deleting extends

Resources

Resources

Want to know more?

- White paper: http://sigops.org/sosp/sosp11/current/2011-Cascais/printable/11-calder.pdf
- Video by Brad Calder: http://www.youtube.com/watch?v=QnYdbQO0yj4
- More slides by Brad Calder: http://sigops.org/sosp/sosp11/current/2011-Cascais/11-calder.pptx
- Blog post by 8Kmiles: http://8kmiles.com/azure-storage-high-level-architecture/
- Erasure coding: http://www.snia.org/sites/default/files2/SDC2012/presentations/Cloud/ChengHuang_Ensuring_Code_in_Windows_Azure.pv3.pdf

Q&A