SDC 18 SNIA EMEA

FEBRUARY 2018
TEL AVIV, ISRAEL

STORAGE DEVELOPER CONFERENCE

Where Network meets Storage

Three ways of integrating Network Protocol into a Storage platform

Mark Rabinovich
Visuality Systems LTD

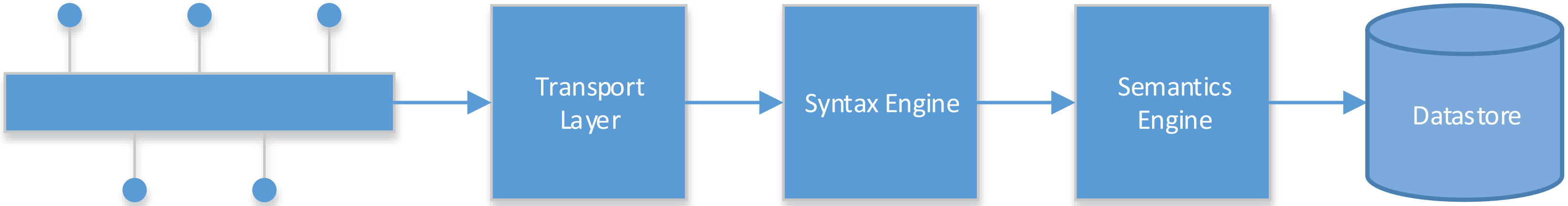
Scope

- Mostly – SMB
    - Shares VS experience in integrating NQ Storage™
- Some considerations applicable to other file sharing methods (NFS).

Challenges

- Integrate file sharing (SMB) into a storage solution
- As seamless as possible
  - The fewer APIs the better
- As generic as possible
  - Good API coverage (contradicts the above)
- Trade-off between functionality and flexibility
- Trade-off between scalability and flexibility

Architectural view



- Transport
  - Accepting connections
  - Delivering requests
  - Transmitting response
- Syntax
  - Parsing requests
  - Composing responses
- Semantics – states
  - Connections
  - Open files
  - Etc.
- Datastore
  - Files and directories

2018 Storage Developer Conference EMEA. © Visuality Systems LTD. All Rights Reserved.

Constraints

- Minimize latencies
- Context switches may be painful
    - Avoid ?
    - Minimize ?
    - Decrease overhead ?
- Where it happens?
    - Between Transport and Syntax
    - Between Syntax and Semantics
    - Inside
        - Critical sections – shares state

Low-latency solutions

- User-space networking
  - DPDK
- Light-weight threads
- Non-preemptive threading

Transports

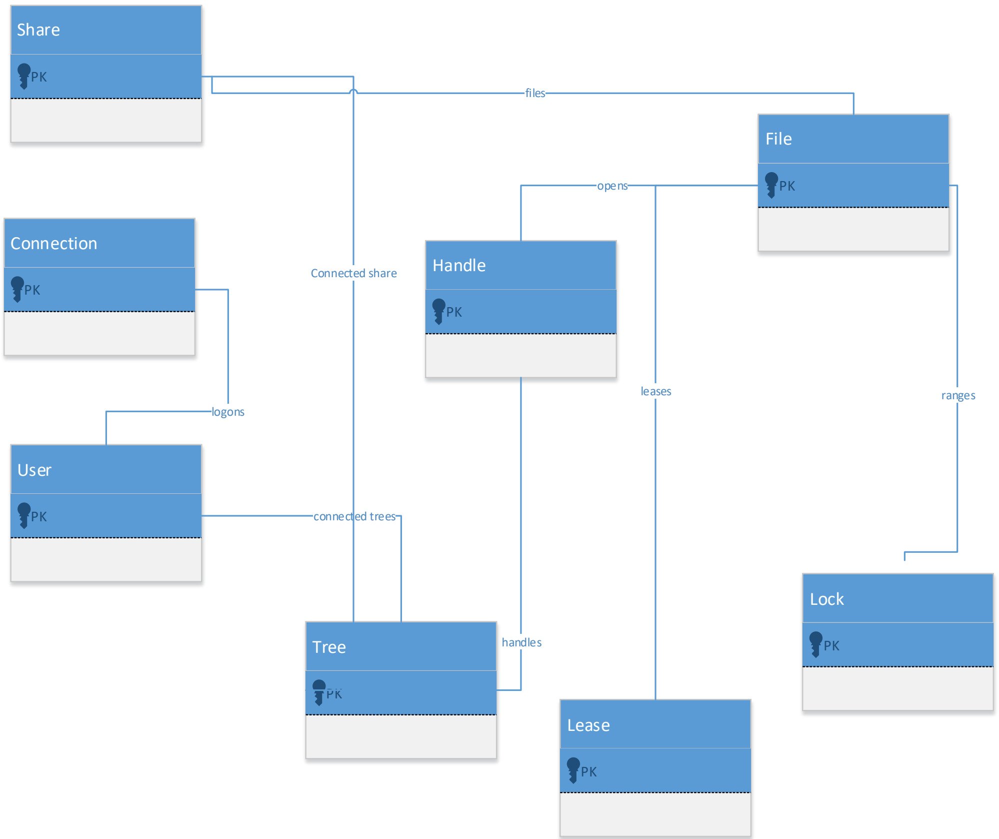
- Sockets
  - Significant latencies
  - Easy to implement
- RDMA/SMBD
  - Low latencies
  - Expensive
- User-space solutions
  - DPDK
  - Does not couple with TCP – needs both sides (as RDMA)

Syntax

- Process a request
  - Receive from Transport
  - Parse
  - Delegate
- Compose a response
  - Receive data and metadata from Semantics
  - Transmit through Transport

Semantics

- Strictly speaking some of the state (Connection, User, Tree) does not belong to file semantics
- Where to handle the above entities?



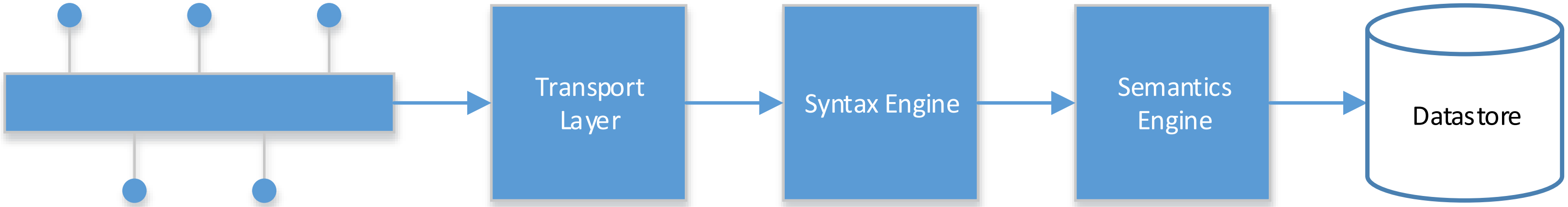
Relationships

- This is not about multithreading but rather about code dependencies
- Syntax to Transport – one-to-many. Multiple transports may be plugged (e.g. BSD sockets + SMBD).
- Syntax to Semantics. One-to-many:
  - NTFS semantics
  - IPC semantics (RPCs)
  - Printing semantics

Clustering

- Syntax is per-node
- Semantics is (partially) cross-node
  - Handles (only persistent)
  - Range locks
  - Leases
- Dedicated replication vs common replication (as in CTDB):
  - Dedicated replication grants better performance
  - Common replication is less expensive (both in terms of development efforts and maintenance efforts).

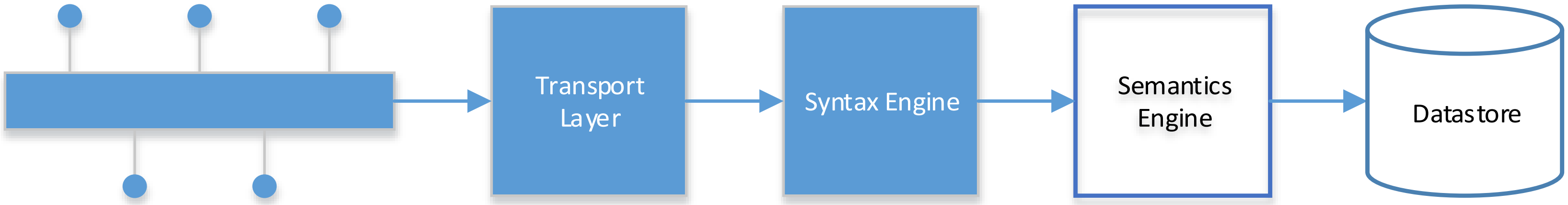
Method – Transport, Syntax, Semantics



- ❐ The best fit for a standalone NAS
- ❐ Clustering (if any) must be internal
- ❐ Semi-dedicated replication (persistent handles, locks, leases)
- ❐ Cross-protocol may be tricky

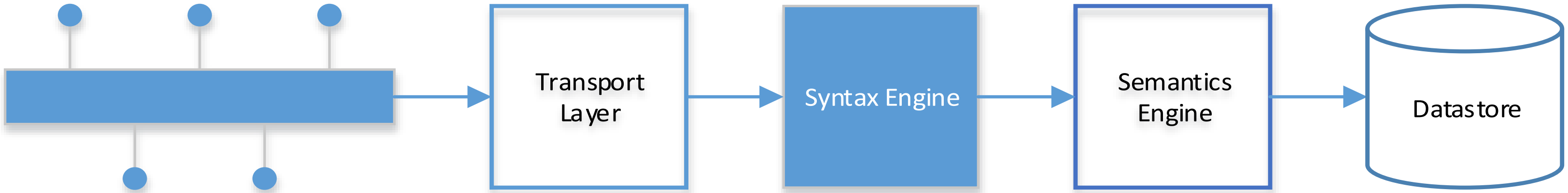
2018 Storage Developer Conference EMEA. © Visuality Systems LTD. All Rights Reserved.

Method – Transport, Syntax



- For clustered storage
- Clustering is external. Replication is out of the scope
- Some state remains inside (connections, users, trees) and is not replicated
- Some user-space solutions may be applied

Method – Syntax only



- ❒ For clustered storage
- ❒ For high-end storage
- ❒ For high scalability
- ❒ Clustering is external
- ❒ User-space solutions may be easily applied.

2018 Storage Developer Conference EMEA. © Visuality Systems LTD. All Rights Reserved.

Method Comparison

Method	User-space solutions	Replication (if at all)	Performance	Scalability
Transport Syntax Semantics	(almost) not available	Inside	Basic	High
Transport Syntax	some available	Outside	Good	High
Syntax only	available	Outside	The best	Even higher

2018 Storage Developer Conference EMEA. © Visuality Systems LTD. All Rights Reserved.

Thank you

www.visualitynq.com

markr@visualitynq.com