SNIA ESF | ETHERNET STORAGE

Transactional Models & Storage Requirements

Live Webcast
October 31, 2017
10:00 am PT

SNIA Legal Notice

SNIA ESF | ETHERNET STORAGE

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Today's Presenters

Alex McDonald
NetApp

J Metz
Cisco

SNIA at a glance

SNIA ESF | ETHERNET STORAGE

160
unique member companies

3,500
active contributing members

50,000
IT end users & storage pros worldwide

Learn more: **snia.org/technical**

@SNIA

Transactional Models & Storage Requirements

- Transactions; a (briefish) history
- Current transactional systems
- The future of transactions
- Transactions and storage; a summary

This is a big subject, and in our limited time today, we're only going to cover some of the topics. If you want a more in depth discussion on anything we've covered (or missed), use the question area and we'll tackle it in the post session blog at ***http://sniaesfblog.org/***

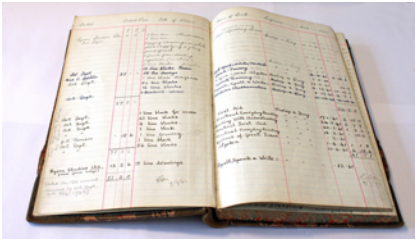
What is a transaction?

- "An input message to a computer system dealt with as a single unit of work"
  - Not terribly satisfactory…
- A piece of information or an action that moves a system from one state to another
  - The "system" maintains at least the current state (and very often the original state and the action too)
  - Persistent; existing over a prolonged period
  - Durable; able to withstand wear or damage
- Examples:
  - J Metz transferred $10 from his bank account to mine
  - Carol booked an airline ticket and reserved seat 17A
  - Kirsten borrowed a book from Trinity College Library, Dublin
  - Fiona put up a new picture on Facebook

Transactions; a (briefish) history

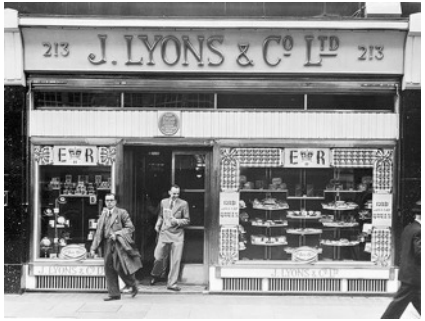
- Recording important events
  - Often of exchange; financial
  - Recorded in **ledgers**
    - What, Why, When, How, Where and Who
  - Ledger medium was generally permanent storage
    - Earliest examples were clay, papyrus
    - Later, but pre-computer; on paper
      (and sometimes still recorded that way)
- We'll extend the idea of a transaction beyond the financial later

Barley rations, Sumerian
circa 2350 BCE



Early computer systems

- Lyons' LEO
  - LEO -- Lyons Electronic Office I
  - UK bakery business
  - First business applications in November 1951
    - Payroll & inventory
  - Storage devices
    - Paper tape, card, 100 line/minute printer, magnetic tape
    - "Ultrasonic delay line memory based on tanks of mercury, with 2K (2048) 35-bit words (i.e., 8¾ kilobytes)"
  - First significant change from paper in 1000s of years



The rise of databases & financial machines

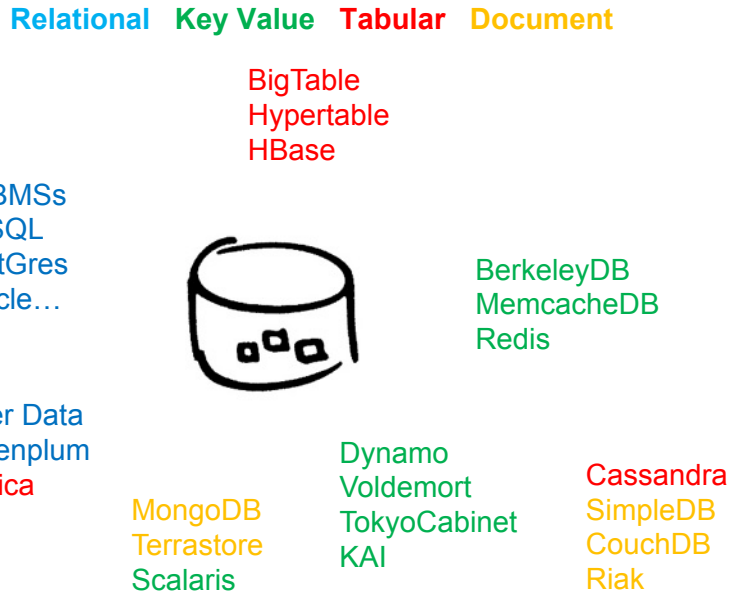
- Banking & financial drove transactional systems
- Paper ledgers converted to databases
- IBM, Burroughs et al
  - Company unique bank clearing systems
  - Eg IBM IMS; hierarchical DB with transaction processing
  - IBM DB2 & Oracle; relational databases
- DASD devices join tape
  - CKD (Count Key Data)
  - Fixed block
- People:
  - Jim Gray, Turing Award "for seminal contributions to database and transaction processing research"
  - Codd, Date: relational database theory

Agenda

- Transactions; a (briefish) history
- **Current transactional systems**
- The future of transactions
- Transactions and storage; a summary

Database types

- Relational
  - Oracle, MySQL, PostGres
- Key Value
  - MemcacheDB, BerkeleyDB
- Tabular
  - Cassandra, BigTable
- Document
  - CouchDB, Riak, MongoDB
- Types overlap; useful if loose classification



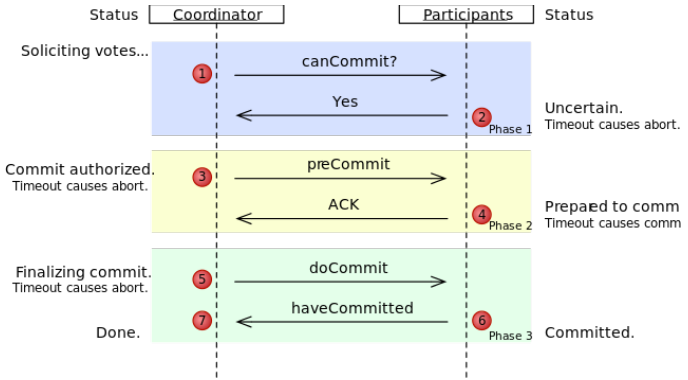
Transactional ACIDity

- Atomicity
  - Transactions are all or nothing; if part fails, the whole fails and state is left unchanged
- Consistency
  - Transactions take the system from one valid state to another; it's not possible for an application to experience invalid states
- Isolation
  - Concurrent execution of transactions (i.e. in parallel) results in a state that would be obtained if transactions were executed sequentially (i.e. serially)
- Durability
  - Once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.

No, not that kind of acid

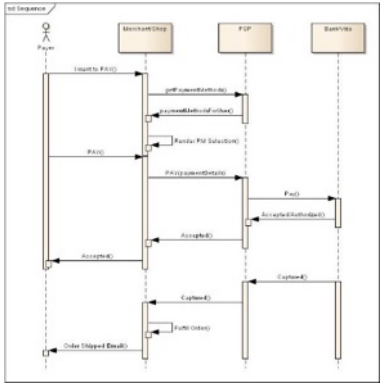
To lock or not to lock…

- 2 phase commit
  - Co-ordinator & cohorts
  - Voting system; all YES then COMMIT
  - Otherwise rollback
- Locking & 2PC provides consistency
  - But they're expensive
  - Hard to make ACID when
    - Distributed transactions across multiple geographic nodes; transactions with high latencies
    - Humans do the "locking"



Online web transactions

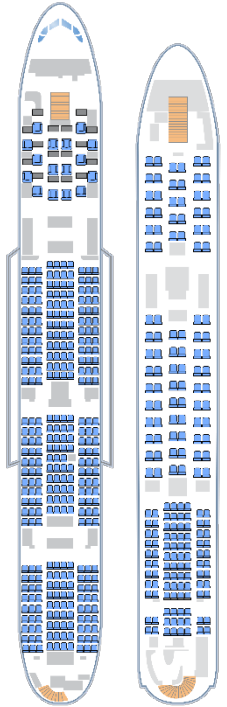
- Example: Web based payment systems
  - Online web purchase by card
  - Vendor website interfaces with credit card application or Payment Service Provider
  - Uses 2 phase commit to ensure debit & credit side either both happen; or neither happens
  - Applications as lock managers



Airlines & blocking seats

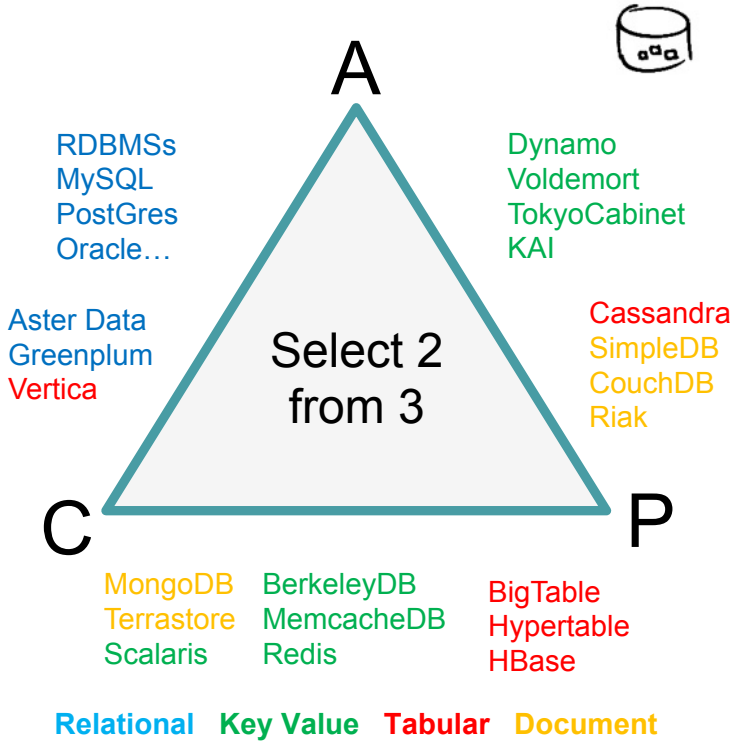
SNIA ESF | ETHERNET STORAGE

- Example: Airline seat booking system
  - Reserve a seat
  - Most times; it works
  - Sometimes it doesn't, so do it again
  - Customers as resource managers, since there's no locking



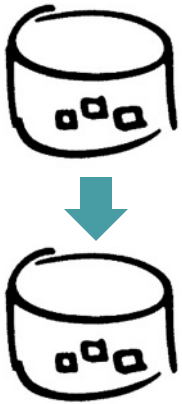
The imperfect world: CAP theorem

- CAP
  - C is for Consistency: all clients have the same view
  - A is for Availability: clients can always read and write
  - P is for Partition tolerance: bits may break, but the whole continues to work
- Select 2 from 3



How storage helps

- ACIDity
  - Durability
    - Committed data to disk, guaranteed
    - Persistency of data; includes sophisticated mirroring solutions
  - Snapshots, clones
    - Automatic or application requested
  - Consistency points
    - Systems move from valid state to valid state
  - Covers from cache all the way out to SSD or traditional disk
- Plus
  - Encryption, compression, deduplication, …

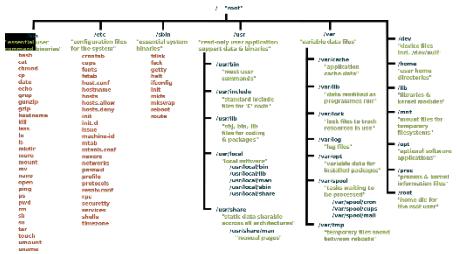


How storage helps

- Block based for traditional high volume low latency transactions
  - aka SAN
  - Programmatic interfaces into storage subsystem
  - Example: Oracle quiesce and snapshot for backup

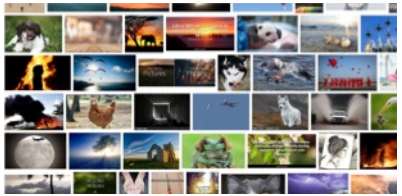
How storage helps

- File based for variety of solutions
  - aka NAS
  - Provides directory, file and byte level locking
  - Includes SQL databases, document & object-type stores
  - IoT specific filesystems & protocols



How storage helps

- Object drives & systems
  - Key value support
  - “Smart” object drives that support metadata storage and query
  - Support for the “new transaction”
    - For example; photographs, video, IoT type data

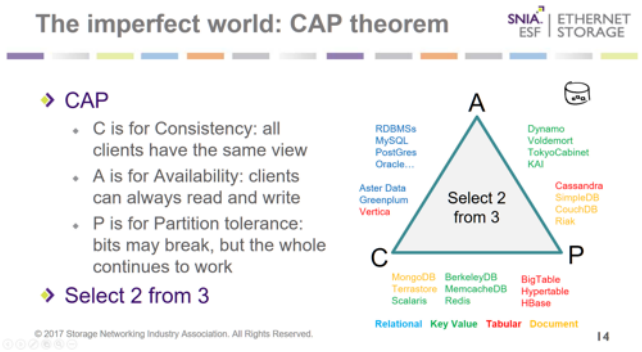


Agenda

- Transactions; a (briefish) history
- Current transactional systems
- **The future of transactions**
- Transactions and storage; a summary

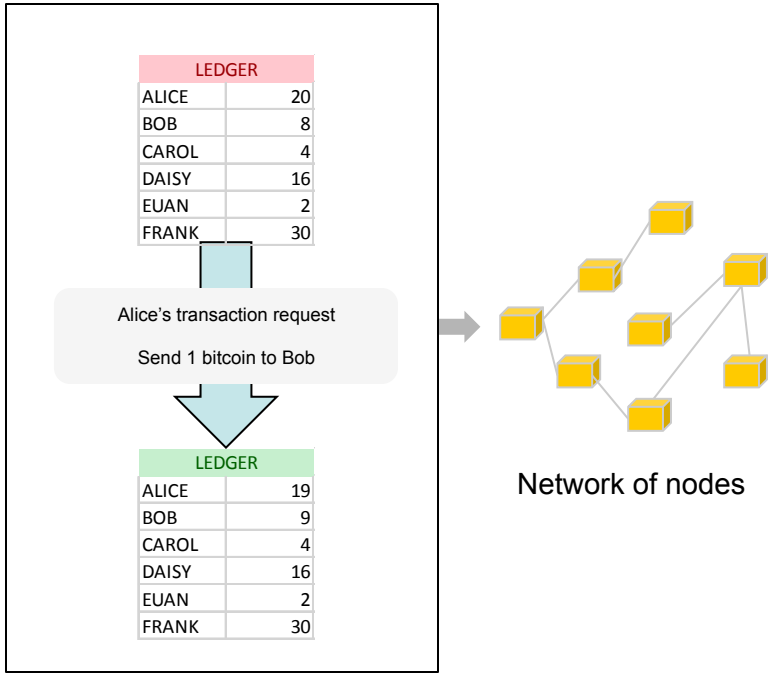
The future of transactions

- The imperfect world of transactions
- Does blockchain provide relief?
- bitcoin is the poster child for blockchain
  - Cryptographically secured ledgers
  - Cryptoledgers can have executable event-triggered actions



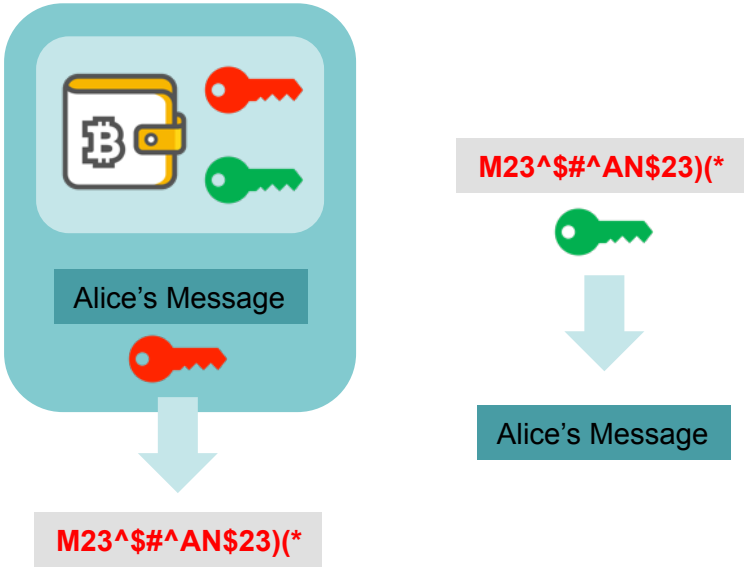
The blockchain ledger

- Decentralized ledger on multiple nodes; each has a copy of the entire chain of ledger entries
- Alice sends 1 bitcoin to Bob
  - Send a message to the network; Alice -1 bitcoin and Bob +1
  - Each node applies the transaction to their copy



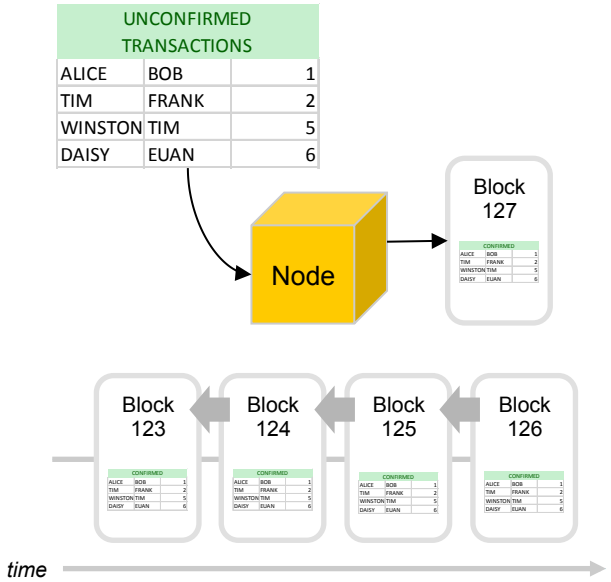
Wallets and keys

- Wallet required to store bitcoins
- Protected by a private and a public key
  - message + private key = encrypted message + public key = message
  - message + public key = encrypted message + private key = message
- Alice sends request encrypted with her private key
- Node decrypts & verifies by using public key to get original request
- Key encryption is effectively a digital signature for the message



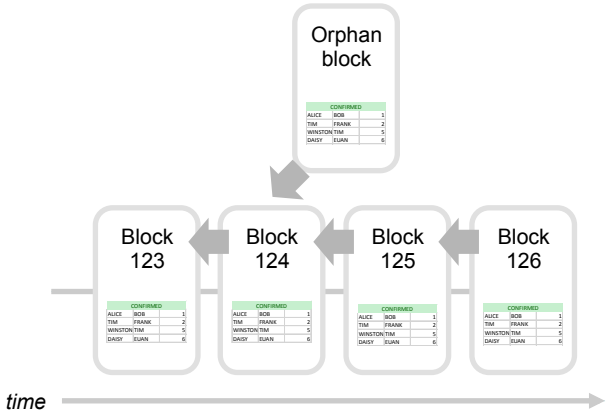
The blockchain ordering

- How do we temporally order transactions?
  - For example, payment then shipment is normal; but not the other way round
  - Can't use timestamps
- Transactions are ordered by grouping into blocks
  - Fixed number of transactions
  - Link to the previous block (hence the name "blockchain")
  - All transactions in specific block are considered to have happened at the same time
  - Transactions not yet in a block are unconfirmed



The blockchain cryptohash

- Node then transmits this as the recommended next new block for the entire system
- But any node can recommend a new block… so which one of many is really next?
- Each block must contain the solution to an **irreversible** cryptographic hash function
  - Content of previous block + random guesses = solution
  - Then broadcasts new block to the whole network
  - Simultaneous solution (which is unusual); nodes must build on longest chain
- System is always consistent



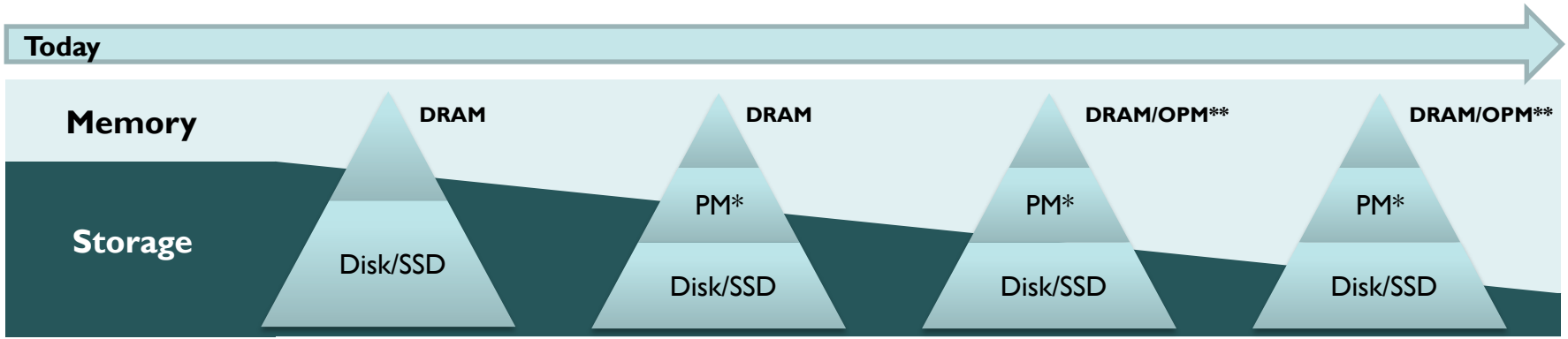
Blockchain and storage

- Persistent memory (PM)
  - Byte addressable storage
- PM is very high bandwidth & low latency
- Guaranteed durability
- Allows entire blockchain in memory for node processing
  - Example: bitcoin ledger only contains transactions, not totals
  - “Balances” require reading all the ledger entries

Memory & Storage Convergence

SNIA ESF | ETHERNET STORAGE

- Volatile and non-volatile technologies are continuing to converge



*PM = Persistent Memory

**OPM = On-Package Memory

New and Emerging Memory Technologies

HMC	3DXPoint™ Memory	Low Latency NAND
HBM	MRAM	
RRAM	PCM	Managed DRAM

Source: Gen-Z Consortium 2016

Agenda

- Transactions; a (briefish) history
- Current transactional systems
- The future of transactions
- **Transactions and storage; a summary**

Transactions and storage; a summary

SNIA ESF | ETHERNET STORAGE

- Changing role of storage
  - Active rather than passive medium for recording information
  - No single storage solution; not all systems are equal
- Future includes movement up the stack into memory and processor domain
  - Persistent memory (PM); byte addressable storage
- Storage more “application like”
  - Increasingly difficult to differentiate between storage, compute & application; for example, HCI
  - Smart, persistent, different solutions for different requirements
  - IoT moves storage to the edge
  - New transactional models; blockchain

Additional information

- Jim Gray, Andreas Reuter
  - PowerPoint “The Whirlwind Tour”
    - research.microsoft.com/~gray/WICS_99_TP/01_WhirlwindTour.ppt
  - Covers transaction state of the art circa 2000 CE
- Persistent Memory
  - https://www.snia.org/PM
- An overview of bitcoin & blockchain
  - https://medium.com/@micheledaliessi/how-does-the-blockchain-work-98c8cd01d2ae

After This Webcast

SNIA ESF | ETHERNET STORAGE

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Ethernet Storage Forum (ESF) website and available on-demand at www.snia.org/forums/esf/knowledge/webcasts
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-ESF blog: sniaesfblog.org
- Follow us on Twitter @SNIAESF

Visit the ESF Webcast Library

SNIA ESF | ETHERNET STORAGE

- 40+ vendor-neutral webcasts covering:
  - Intro to Ethernet Networked Storage
  - Block, File and Object Storage
  - RDMA and NVMe over Fabrics
  - Containers
- 2 popular webcast series:
  - Everything You Wanted To Know About Storage But Were Too Proud To Ask
  - Storage Performance Benchmarking
- All on-demand at https://www.snia.org/forums/esf/knowledge/webcasts-topics

SNIA ESF | ETHERNET STORAGE

Thank You