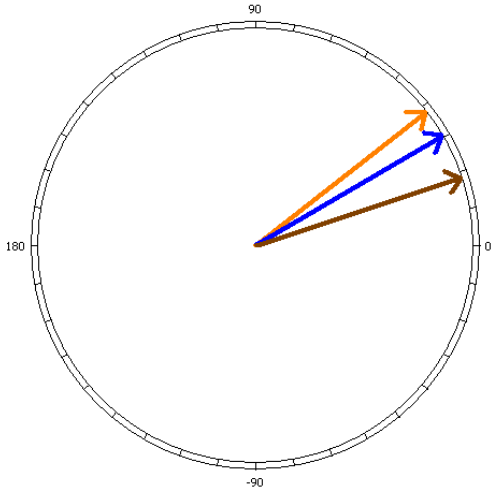
OG&E's SynchroPhasor Historian

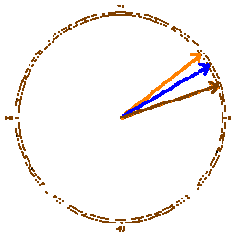


openHistorian
GRID PROTECTION ALLIANCE

Steven E. Chisholm
Oklahoma Gas & Electric
Grid Protection Alliance

OG&E

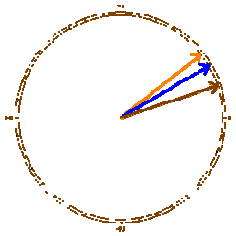
GRID PROTECTION ALLIANCE



Quote

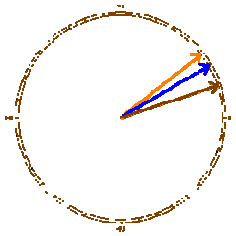
"There's a way to do it better—find it."

-Thomas Edison



OG&E's SynchroPhasor Team

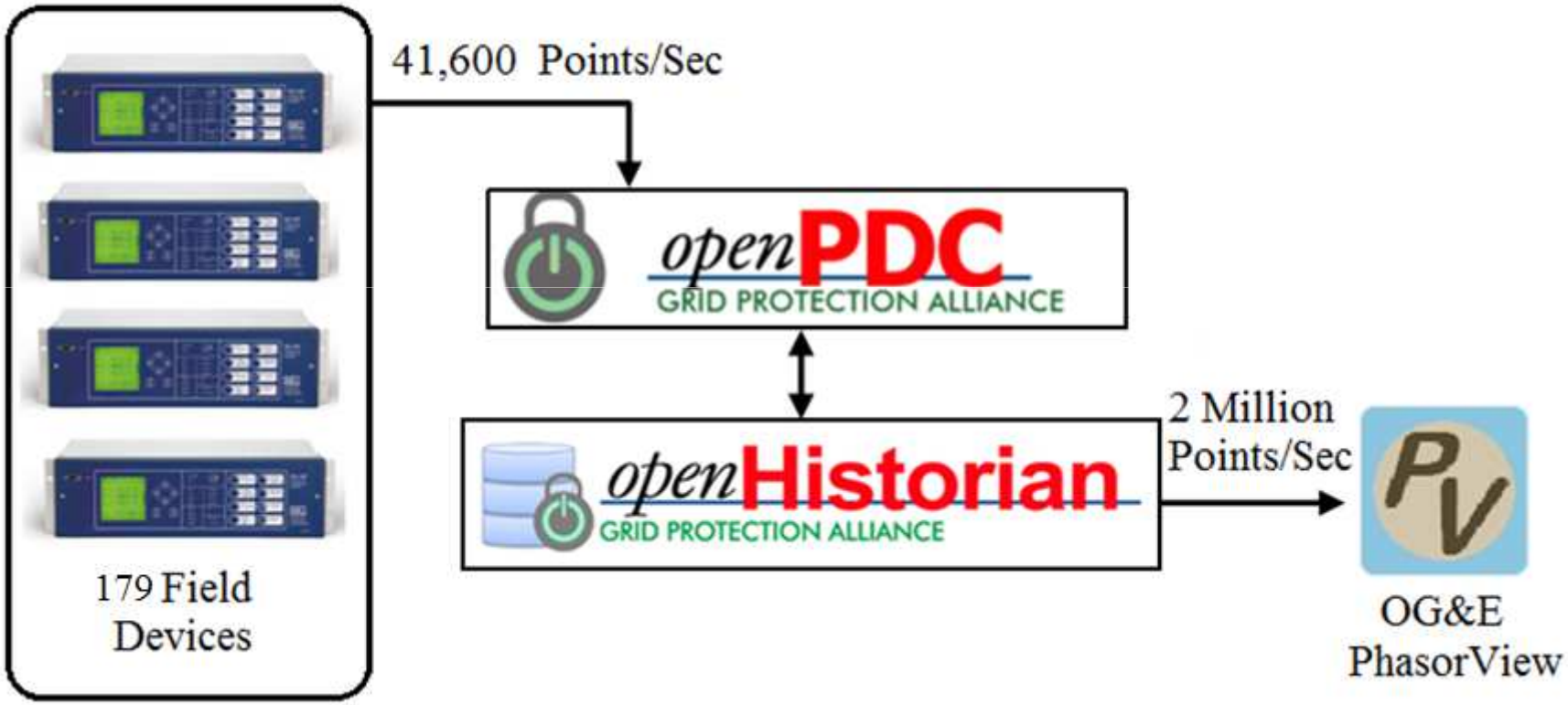
- Steven Chisholm
  - Electrical Engineer
  - 5 Years at OG&E as Power Engineer
  - Self-Taught Programmer (17 years)
  - Application Development for Synchrophasor Technology
- Austin White
  - Electrical Engineer
  - 10 Years at OG&E as Power Engineer
  - Primary User of SynchroPhasor Technology



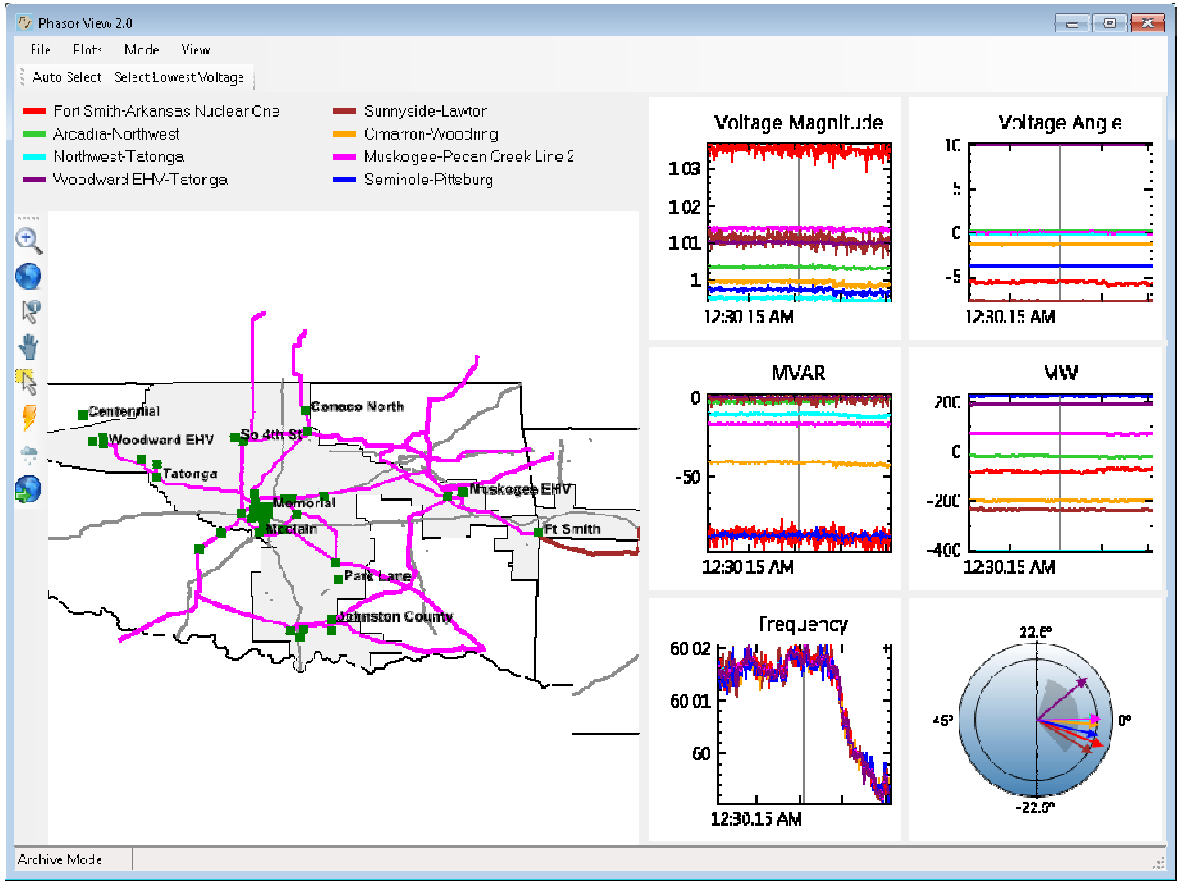
Not Your Typical Historian

- What is an historian? (My Experience)
  - It’s a place where data goes to die.
  - Extracts data in real time
    - Takes 1 week to get 1 week worth of data
- Why is openHistorian different?
  - Crazy fast
  - Easy to use API
  - Services hundreds of clients

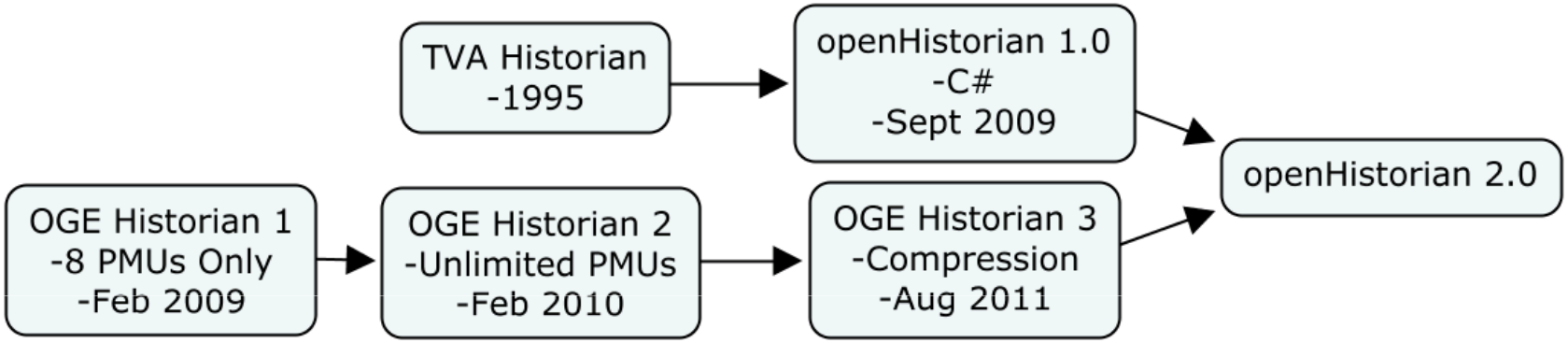
Prospective Architecture



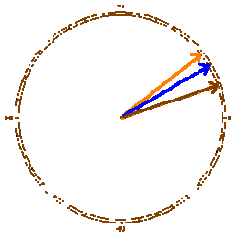
OG&E's PhasorView



History of openHistorian 2.0



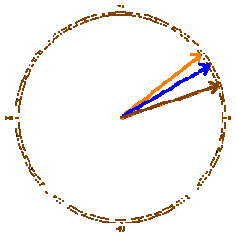
- Built from the ground up taking what we learned from 5 historian products.



Lessons Learned: OG&E Historian 1.0

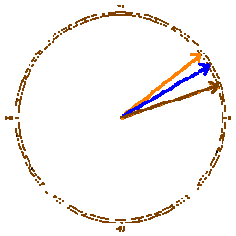
- Structure: MS SQL Table
  - Timestamp,ZoomIndex,Value1,Value2,…,Value56
  - CLUSTER on Timestamp (Crucial)
  - 1 SQL database per day (~600MB)
- Best way to get a subset of the data.
  - Experimented with Database Index on ZoomIndex
    - FAILED - Slow Reads. Slow Writes.
  - Duplicated down sampled data in new table.
  - Full Resolution Table, 1 Per Second Table,

    1 Per Minute Table, 1 per Hour Table.



Lessons Learned: OG&E Historian 2.0

- Structure: MS SQL Table
  - Timestamp, Terminal ID, Zoom Index, Values[8]
  - CLUSTER on Timestamp, Terminal ID
  - 1 SQL database per Month (~700GB)
- Cost for flexibility 50%
- Started storing relay digitals
- Continued down sampled method
- Limited to single threaded queries

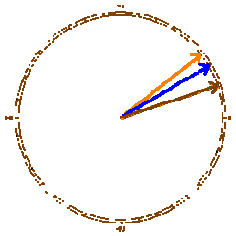


Lessons Learned: OG&E Historian 3.0

- Structure: MS SQL Table
  - Timestamp, Zoom Index, Data[8000]
  - CLUSTER on Timestamp
  - 1 SQL database per Month (~250GB)
- Implemented high speed *lossless* compression
- SQL is slow / SqlDataReader is slow
- SQL won't let data split pages (only 2-3KB BLOBs fit in one 8KB page)

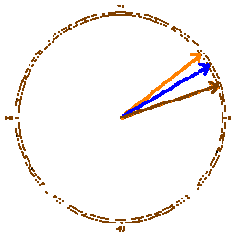
Lossless Defined

"Lossless data compression is a class of data compression algorithms that allows the exact original data to be reconstructed from the compressed data." -Wikipedia



Lessons Learned: GPA openHistorian 1.0

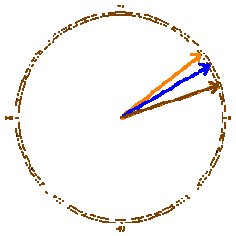
- Structure: File
  - 8KB Block (1 Point): Timestamp, Quality, Value
  - Block Map at end of file
- What We Fixed
  - Expensive Initializing
  - Very High Write Amplification
  - Limited Data Type Support
  - Limited Time Resolution (milliseconds)



Write Speeds

Product	Write (Points/Sec)	2 Trillion Points
OG&E Historian 1.0	5,000	13 Years
OG&E Historian 2.0	300,000	77 Days
OG&E Historian 3.0	1,000,000	23 Days
openHistorian 1.0	70,000	330 Days
openHistorian 2.0	1,000,000 (?)	23 Days (?)

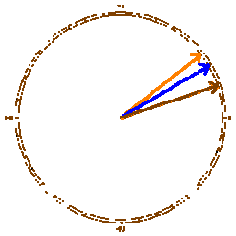
- Writing: Alpha build tests to be ~340,000. Experimental optimizations ~3 million sequential and ~1.4 million random.



Read Speed

Product	Read (Points/Sec)	Queries Per Second	Max Clients Per Server
OG&E Historian 1.0	2,000,000	8.3	33
OG&E Historian 2.0	1,700,000	7.1	28
OG&E Historian 3.0	10,000,000	41.6	166
openHistorian 1.0	1,000,000	4.2	16
openHistorian 2.0	20,000,000 (?)	83.3 (?)	333 (?)

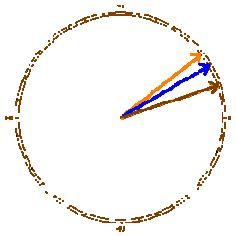
- Assuming: 8 Quantities from 10 PMUs. 3000 samples per Quantity
- Assuming: Quad Core Server. Linear Scaling with CPU
- Reading: Alpha built tests to be ~27 million. Experimental ~47 million.



Database Size

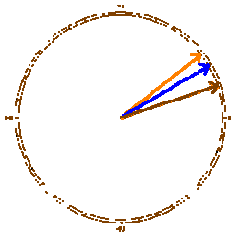
Product	Size (Bytes/Point)	2 Trillion Points
OG&E Historian 1.0	4.4	8.8 TB
OG&E Historian 2.0	6.6	13.2 TB
OG&E Historian 3.0	2.7	5.4 TB
openHistorian 1.0	10	20 TB
openHistorian 2.0	3 (?)	6 TB (?)

- Compression: Been able to get down to 1.8 bytes as a minimum.



openHistorian 2.0: The What

- Data is sorted
- Isolate reads from writes
- Resistant to data corruption
- Transaction like editing
- Out of order inserting
- Low Insertion Lag Time
- ACID Support

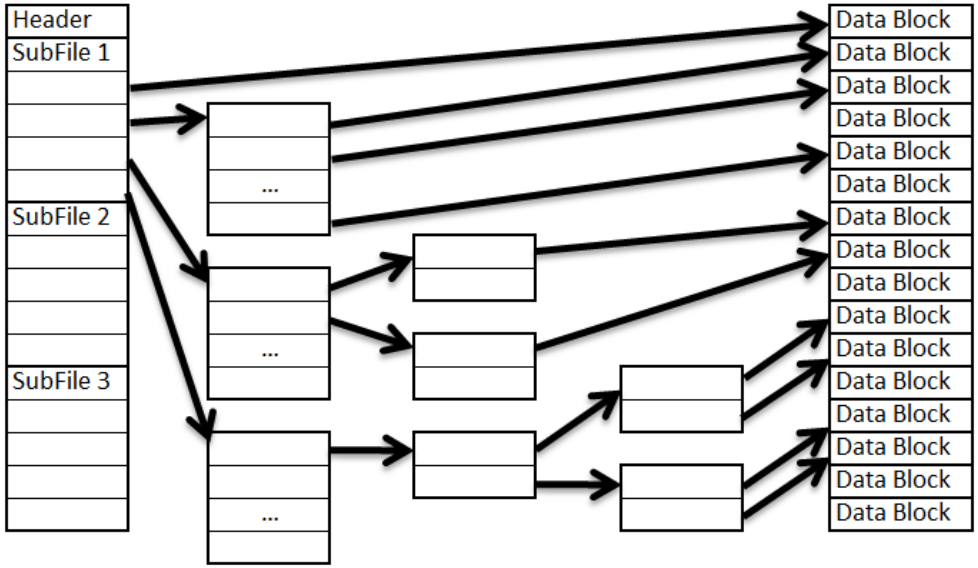


What is ACID?

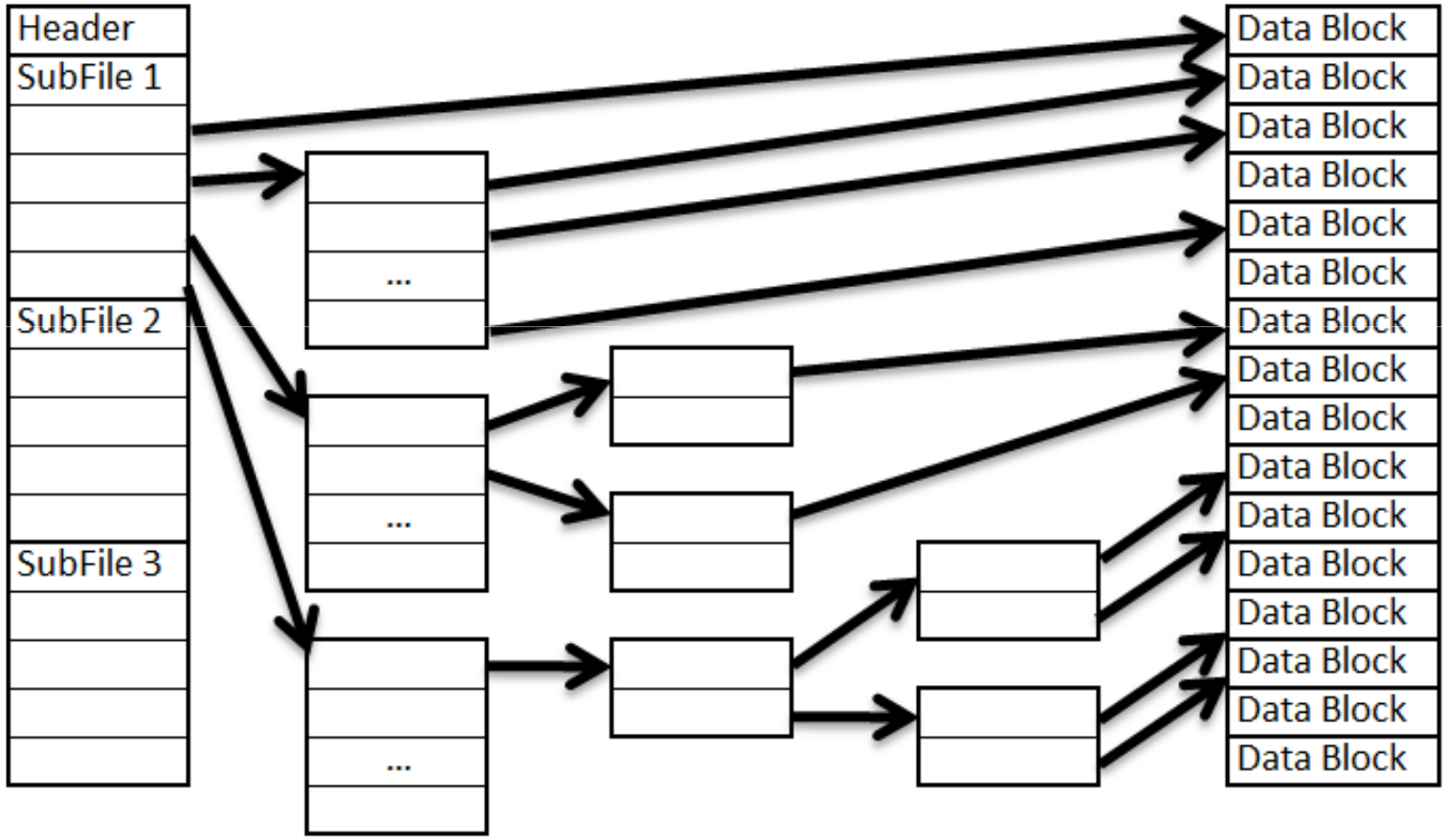
- **Implementing ACID primarily to protect data integrity.**
  - Atomicity - requires that database modifications must follow an "all or nothing" rule. Each transaction is said to be atomic
  - Consistency - ensures that any transaction the database performs will take it from one consistent state to another
  - Isolation - refers to the requirement that no transaction should be able to interfere with another transaction at all
  - Durability - that once a transaction has been committed, it will remain so

openHistorian 2.0: The How

- Wrote a custom virtual file system
  - Supports transactions
  - Read/write isolations through snapshots
  - Resists corruption

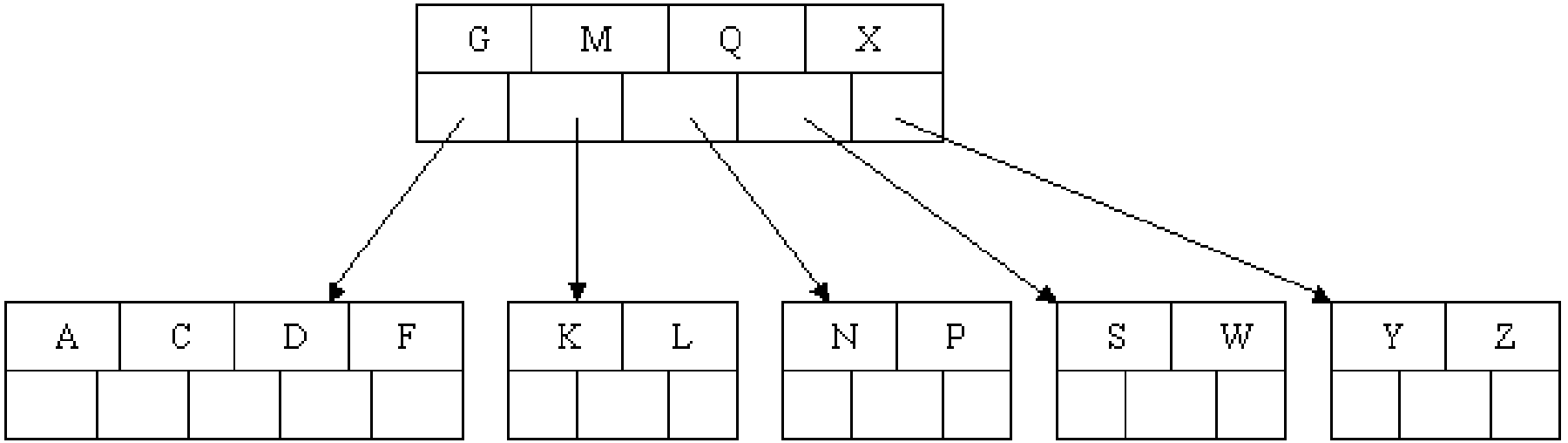


openHistorian 2.0: The How



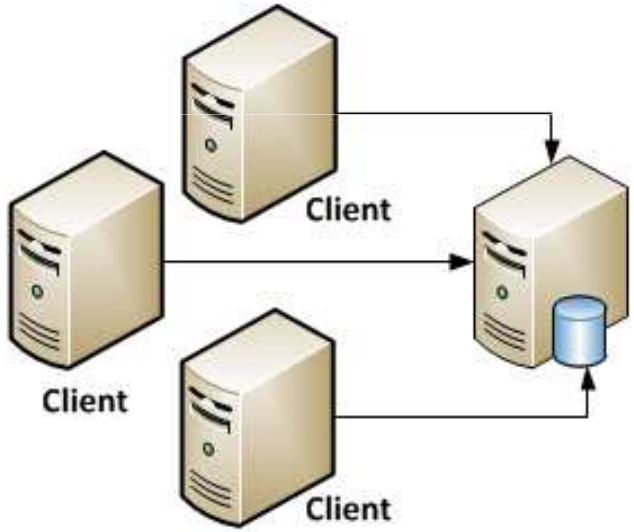
openHistorian 2.0: The How

- Wrote a B+ Tree structure
  - Sorted Data
  - Out of order insertion

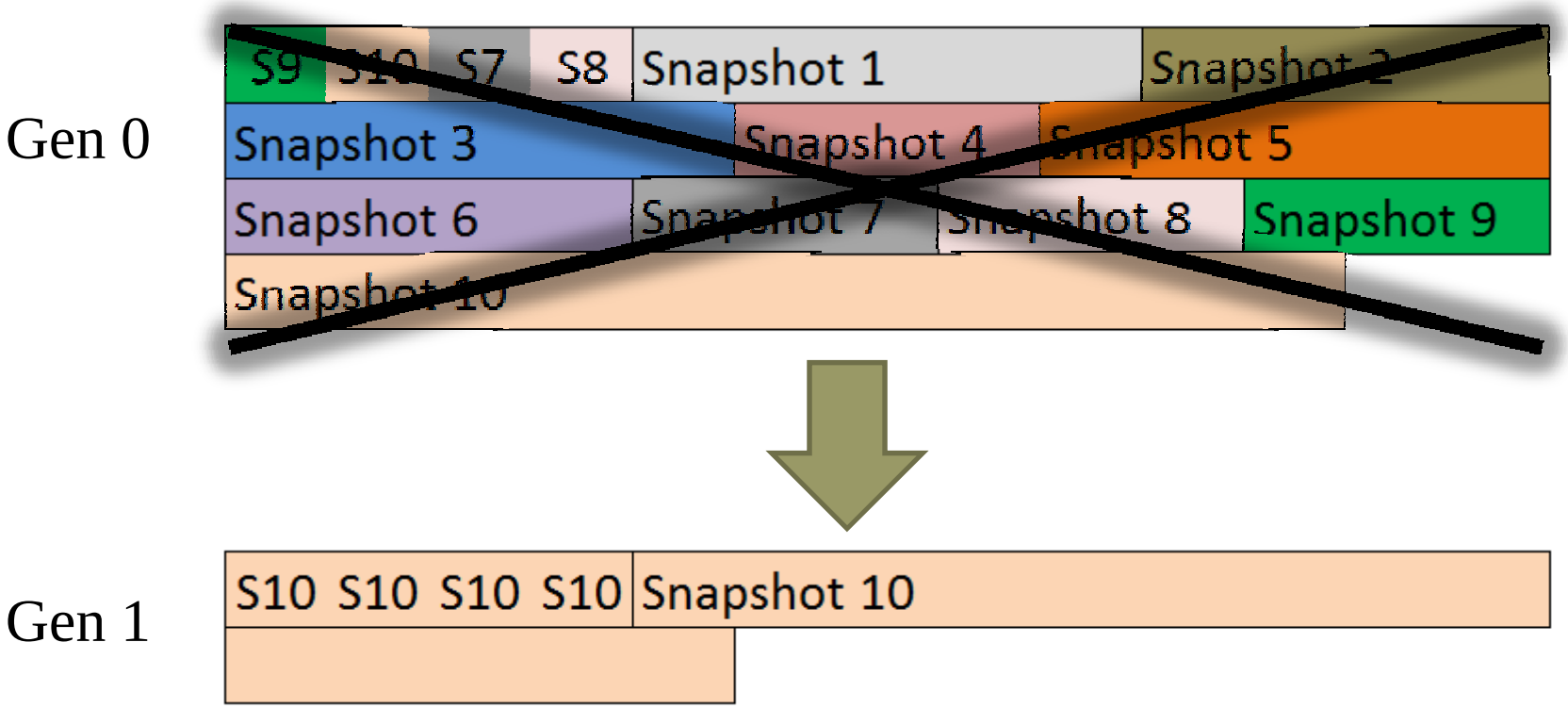


openHistorian 2.0: The How

- Wrote a server management component
  - Instant snapshot of entire database
  - Large memory buffer
    (up to 128GB RAM)
  - Speed
- In Memory Archive File
  - Insert Lag Times <50ms
  - Commit to disk configurable

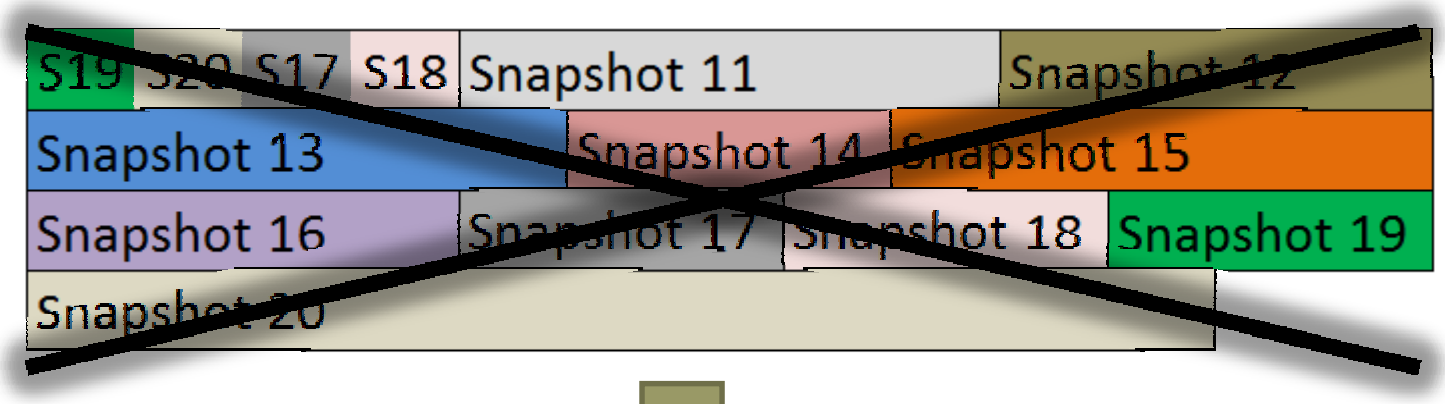


Rollover Architecture

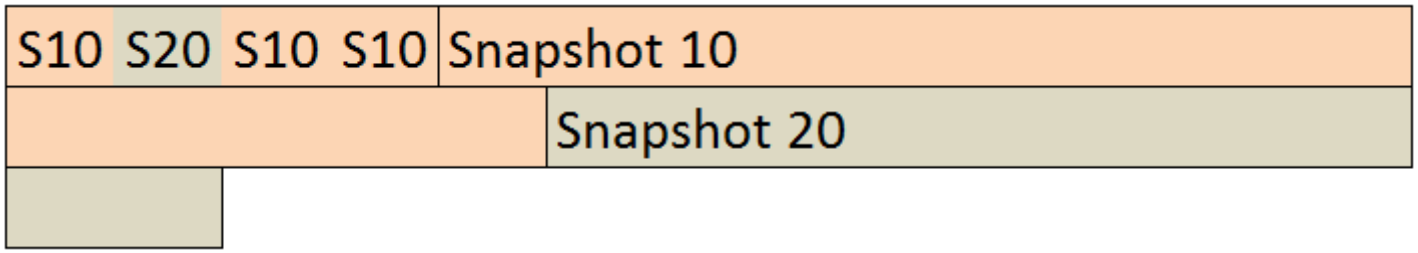


Rollover Architecture

Gen 0



Gen 1



Rollover Architecture

Gen 0

File 1	File 2	File 3	File 4
v. 1	v. 6	v. 11	v. 16
v. 2	v. 7	v. 12	v. 17
v. 3	v. 8	v. 13	v. 18
v. 4	v. 9	v. 14	v. 19
v. 5	v. 10	v. 15	v. 20

Gen 1

File 1	File 2	File 3
v. 5	v. 30	v. 55
v. 10	v. 35	v. 60
v. 15	v. 40	v. 65
v. 20	v. 45	v. 70
v. 25	v. 50	v. 75

Rollover Architecture

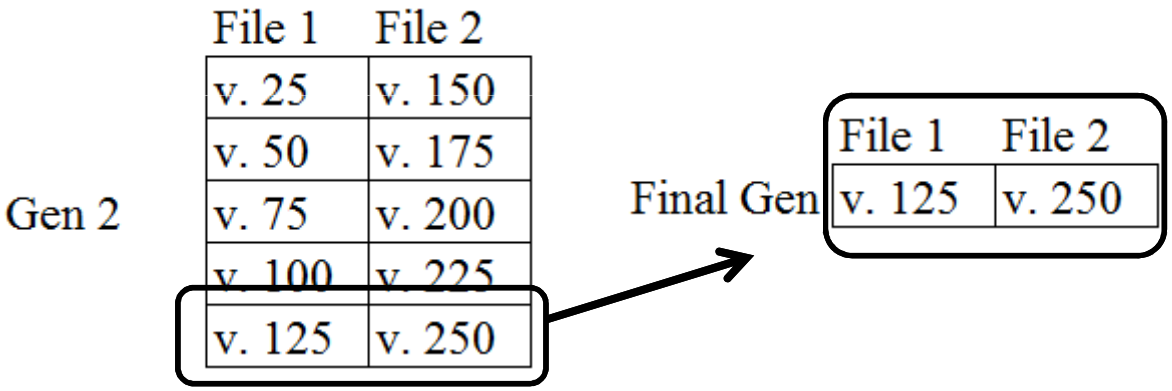
Gen 1

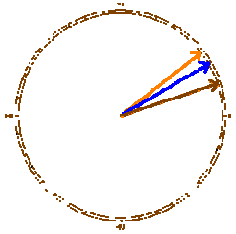
File 1	File 2	File 3
v. 5	v. 30	v. 55
v. 10	v. 35	v. 60
v. 15	v. 40	v. 65
v. 20	v. 45	v. 70
v. 25	v. 50	v. 75

Gen 2

File 1	File 2
v. 25	v. 150
v. 50	v. 175
v. 75	v. 200
v. 100	v. 225
v. 125	v. 250

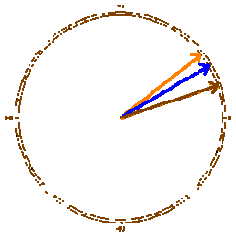
Rollover Architecture





What you configure

- Archive File Writer
  - InMemory, Name=Gen0, NewFile=10MB
- Archive File Manager
  - (Gen0 >> Gen1), NewFile=100MB
  - (Gen1 >> Gen2), NewFile=10GB
  - (Gen2 >> Gen3), NewFile=100GB
- Archive File Retirement
  - Source=Gen3, MaxSize=1TB, Length=1Year

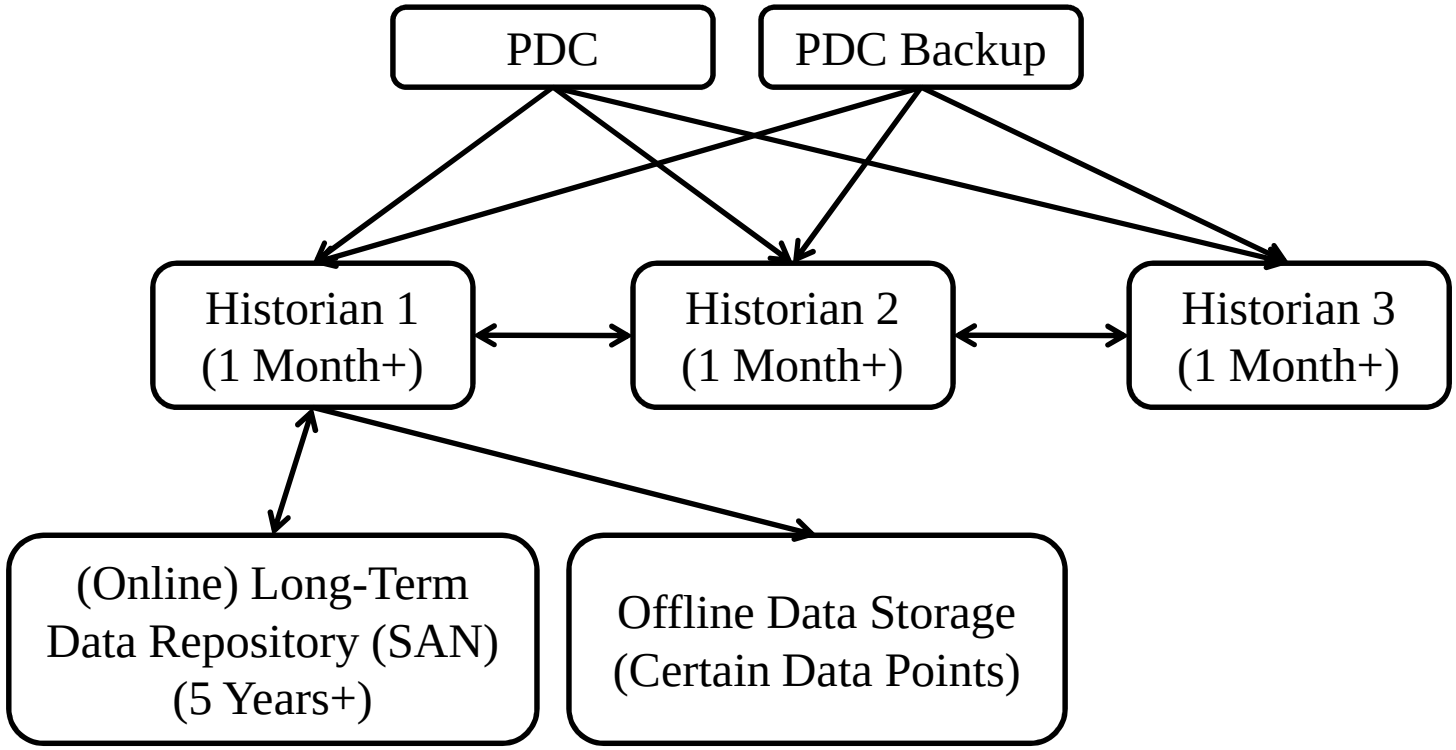


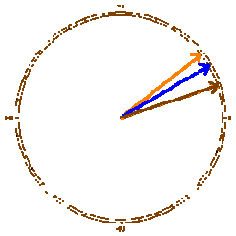
In-Memory Historian

- Archive File Writer
  - InMemory, Name=Gen0, NewFile=1MB
- Archive File Manager
  - InMemory, (Gen0 >> Gen1), NewFile=10MB
  - InMemory, (Gen1 >> Gen2), NewFile=100MB
- Archive File Retirement
  - Source=Gen2, MaxSize=300MB

High Availability/Scaling

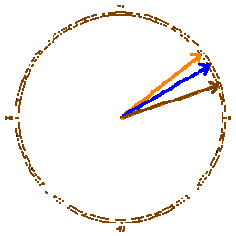
- Supports Clustering / Load Balancing.





openHistorian 2.0: The When

- Source (API Level now)
  - DLL API only
- Alpha Release (3Q 2012)
  - Packaged Install, GUI Tools
- Beta Release (4Q 2012)
  - Compression Added, Stable
- Release (1Q 2013)
  - Bug Free!



Questions?

- Thanks! Feel free to contact us if you have any questions.
  - Steven Chisholm
    - chishose@oge.com (405-553-3764)
  - Austin White
    - whitead@oge.com (405-553-5996)

How you can help

- What are your API requirements?