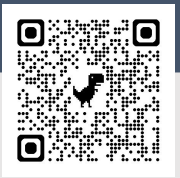
A Roundup of Observability Datastores

Josh Lee • Open Source Advocate @ Altinity • Code Europe

"A Developer will never ask you, 'Hey, what filesystem is that?'"

— Patrick McFadin

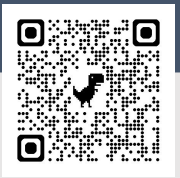


Josh Lee

Open Source Advocate

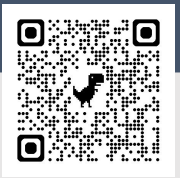
Altinity

Observability = Visibility + Understanding



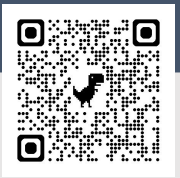
50x

Observability data vs system transaction data



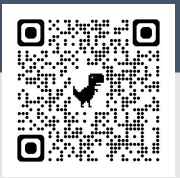
What are we storing?

Metrics, Traces, Logs, Profiles, Events
Labels/Tags
Resource Metadata
Graphs & Topologies
Snapshots & Deltas
Configuration (e.g. alerts, users, dashboards)



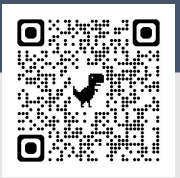
What do we need for observability?

Fast streaming writes
Efficient compression & storage
Time-oriented management
“Real-time” analytics



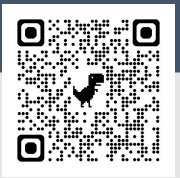
"Anything you can do with a group by, that's what analytics is"

—Peter Marshall



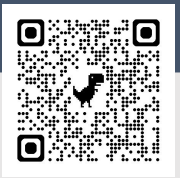
More Requirements

Fast multi-row analytics
Full-text search
Tag/label search
Fast, frequent "last point" reads
Updates?



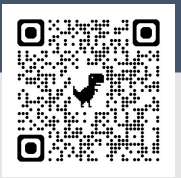
Database Archetypes

OLTP
OLAP
TSDB
Search/Analytics

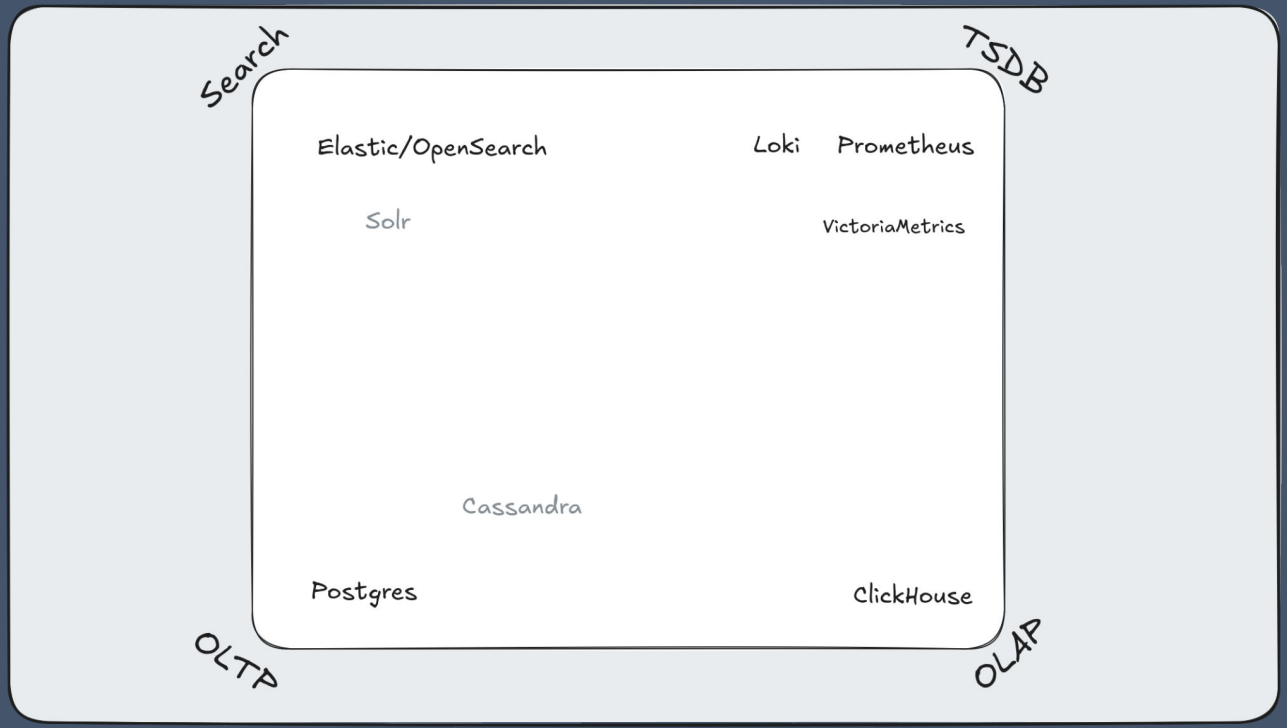


Introducing the cast of characters

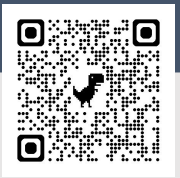
Postgres (OLTP)
Cassandra (OLTP)
Elastic/OpenSearch (Search & Analytics)
Prometheus (TSDB)
ClickHouse (OLAP)



Taxonomies are challenging

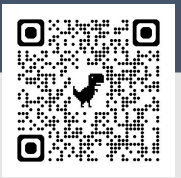


Storage on disk



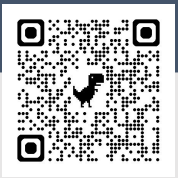
Database Storage Styles

Heap Pages + Commit Log
Time-series Blocks
Parts / Segments

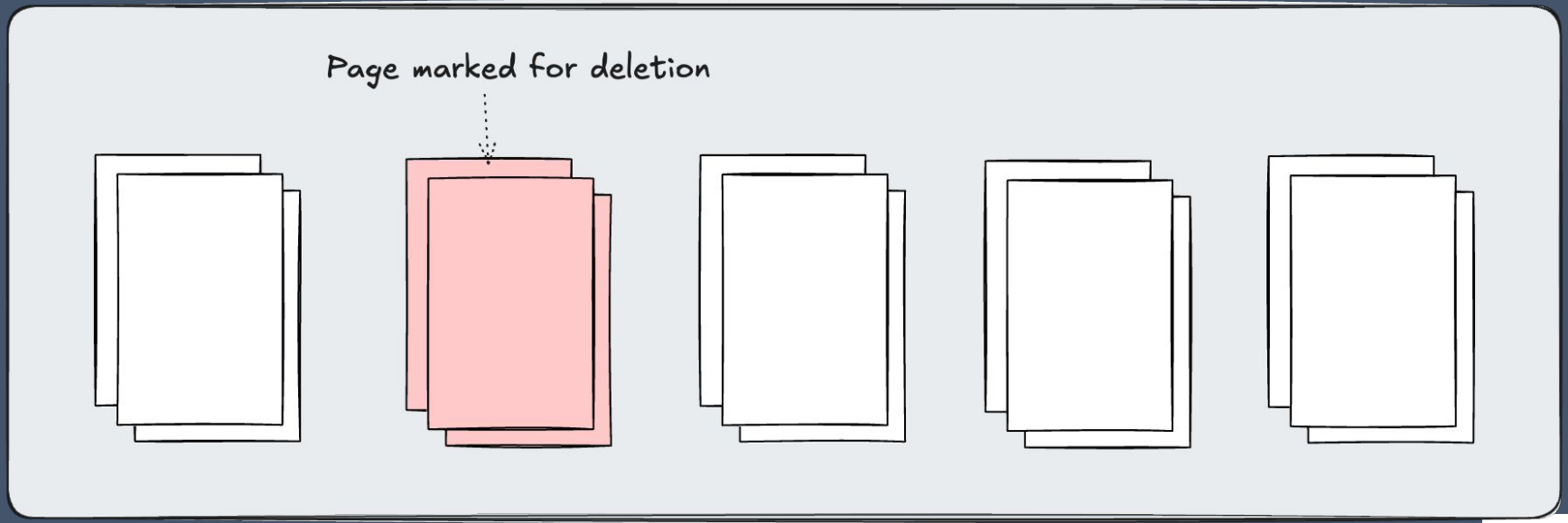


Heap Pages

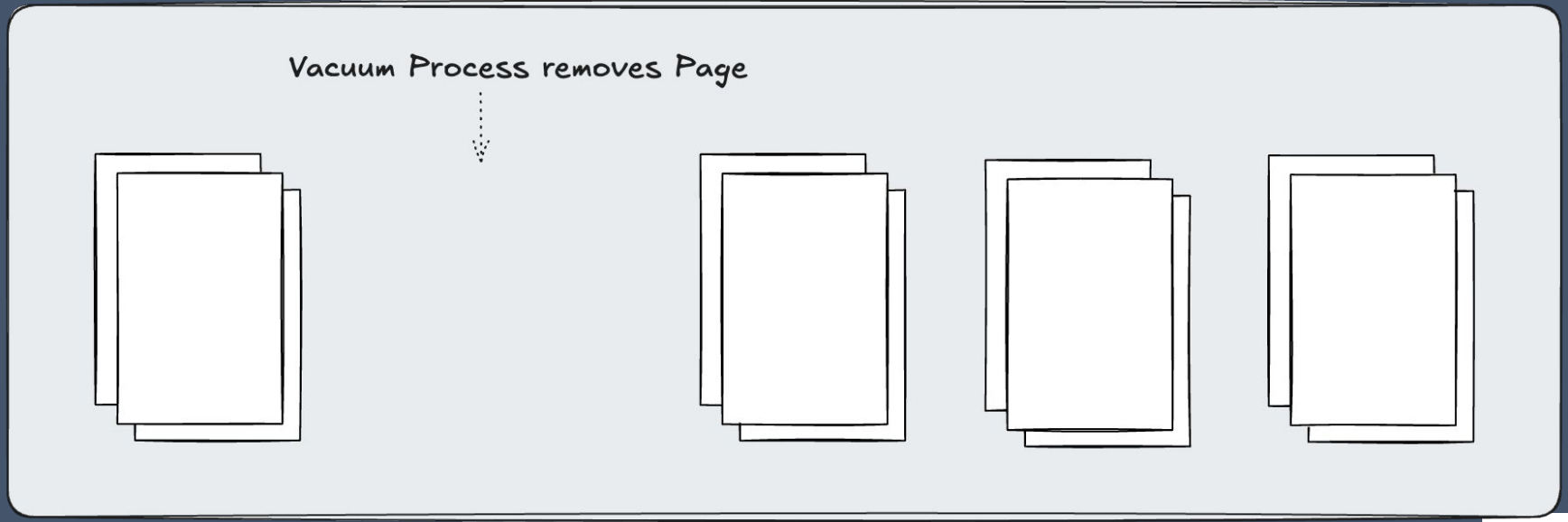
* the JBOD of storage styles



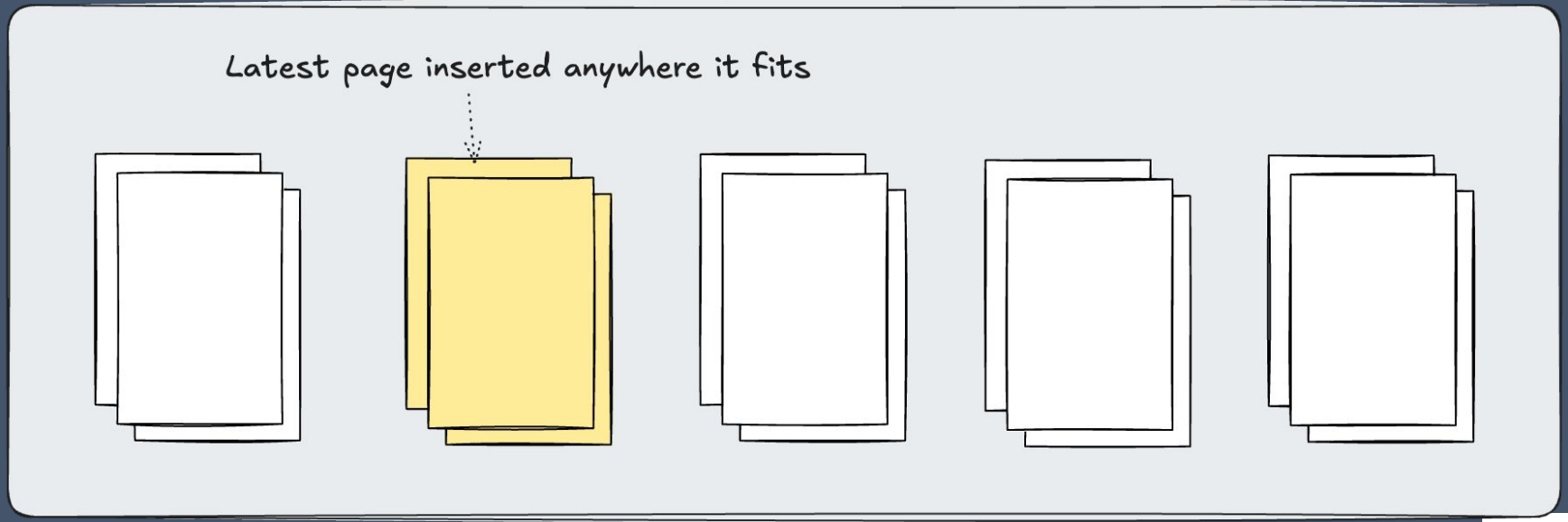
Heap Pages



Heap Pages

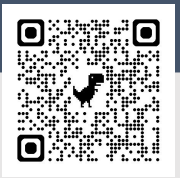


Heap Pages



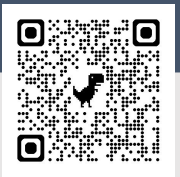
How to make a Postgres

WAL
Heap Pages + MVCC (Multi Version Concurrency Control)
B-Tree Indexes

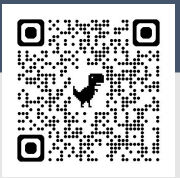


Postgres/MySQL/etc.

Optimized for updates/upserts and row-level reads
Strong ACID guarantees
Scaling horizontally is challenging

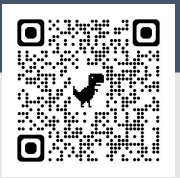


Analytics & Search Architecture



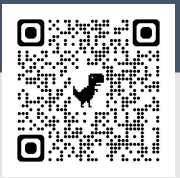
Immutable Parts / Segments

w/ Background Compaction

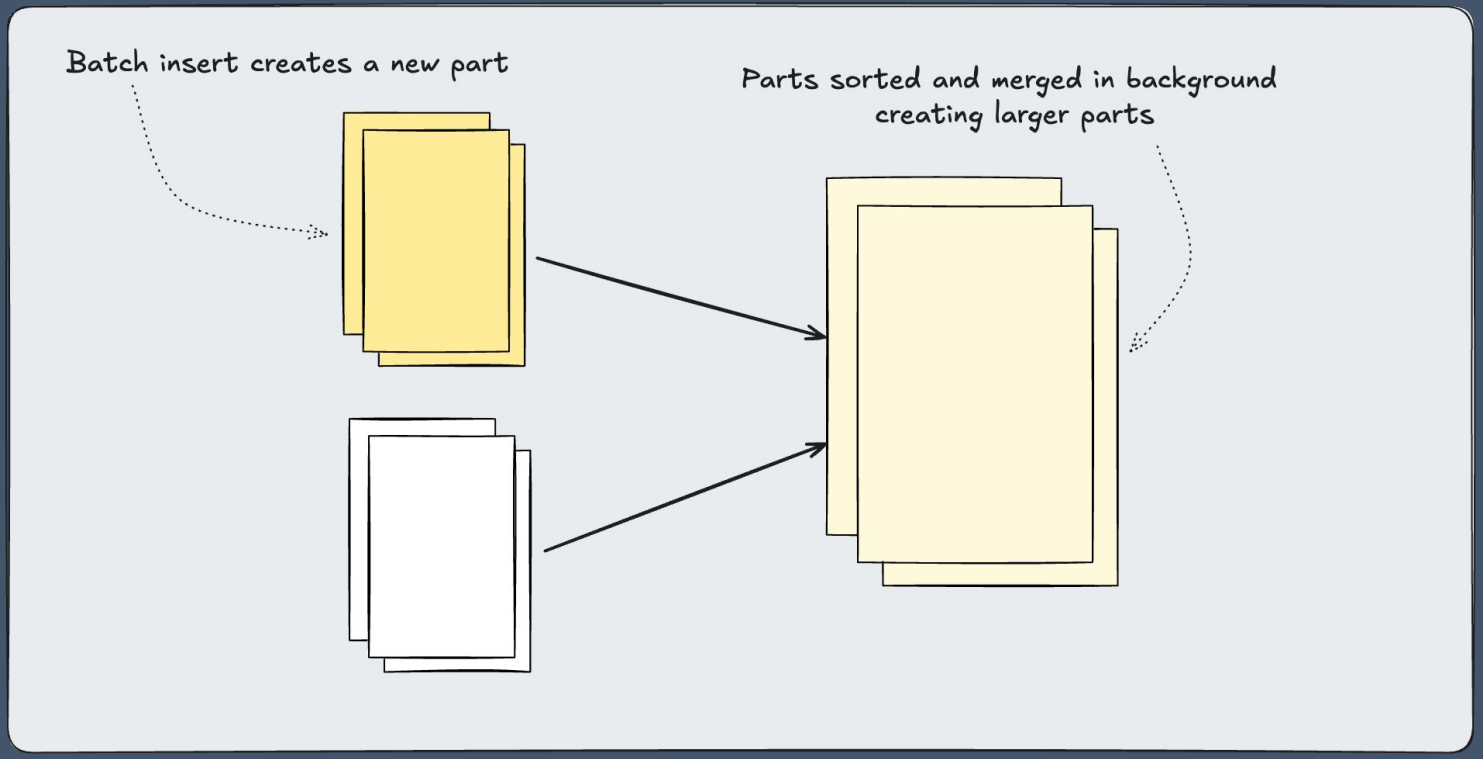


Immutable segment storage:

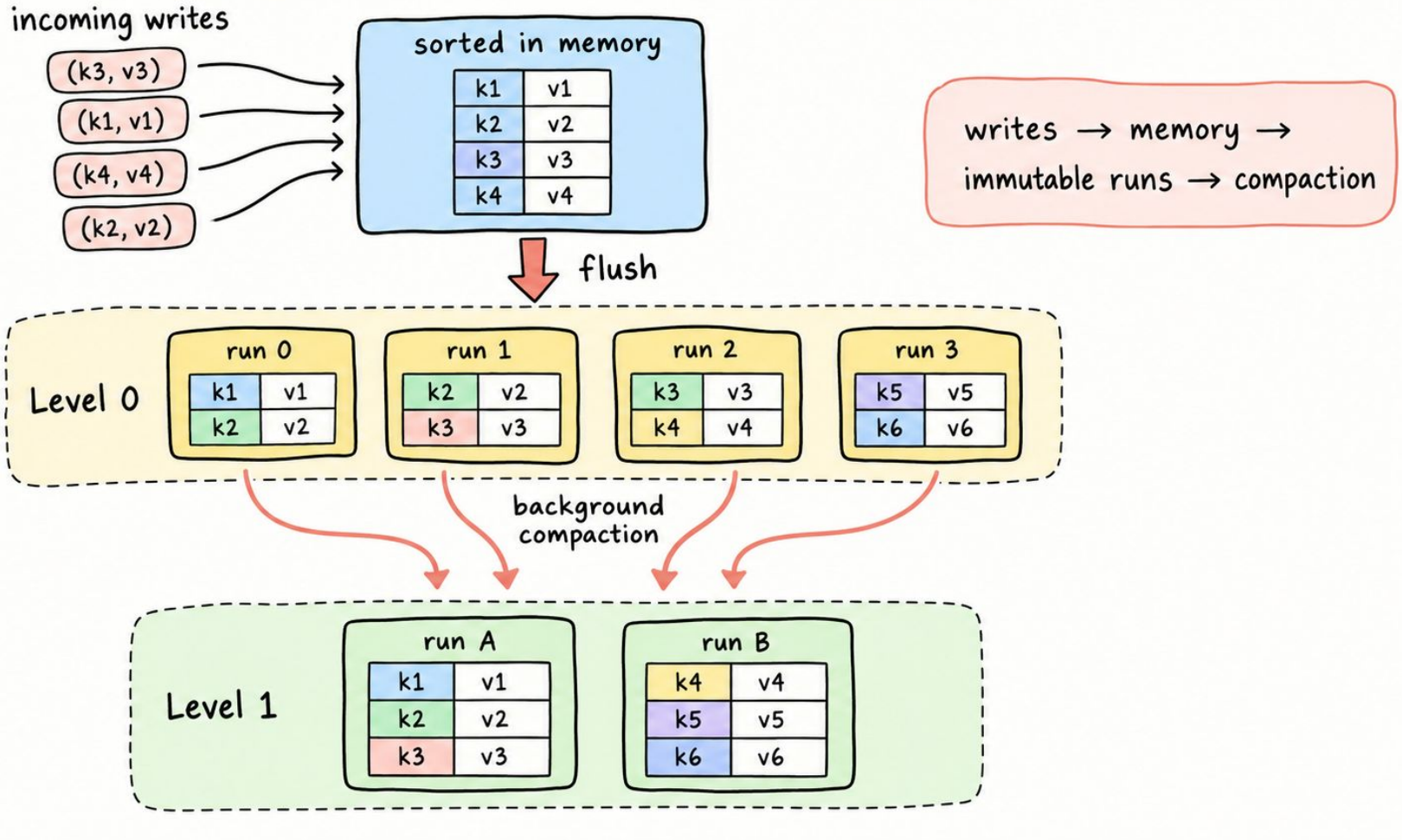
Cassandra, Elastic/OpenSearch, Apache Solr, Splunk, and more…



Immutable Parts / Segments



How an LSM Tree Organizes Writes

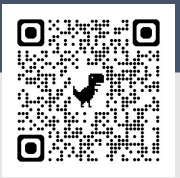


Search & Vector Engines

Inverted Indexes

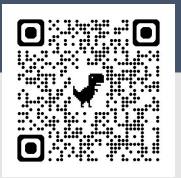
Bloom Filters

Approximate Nearest Neighbor (ANN Graph)



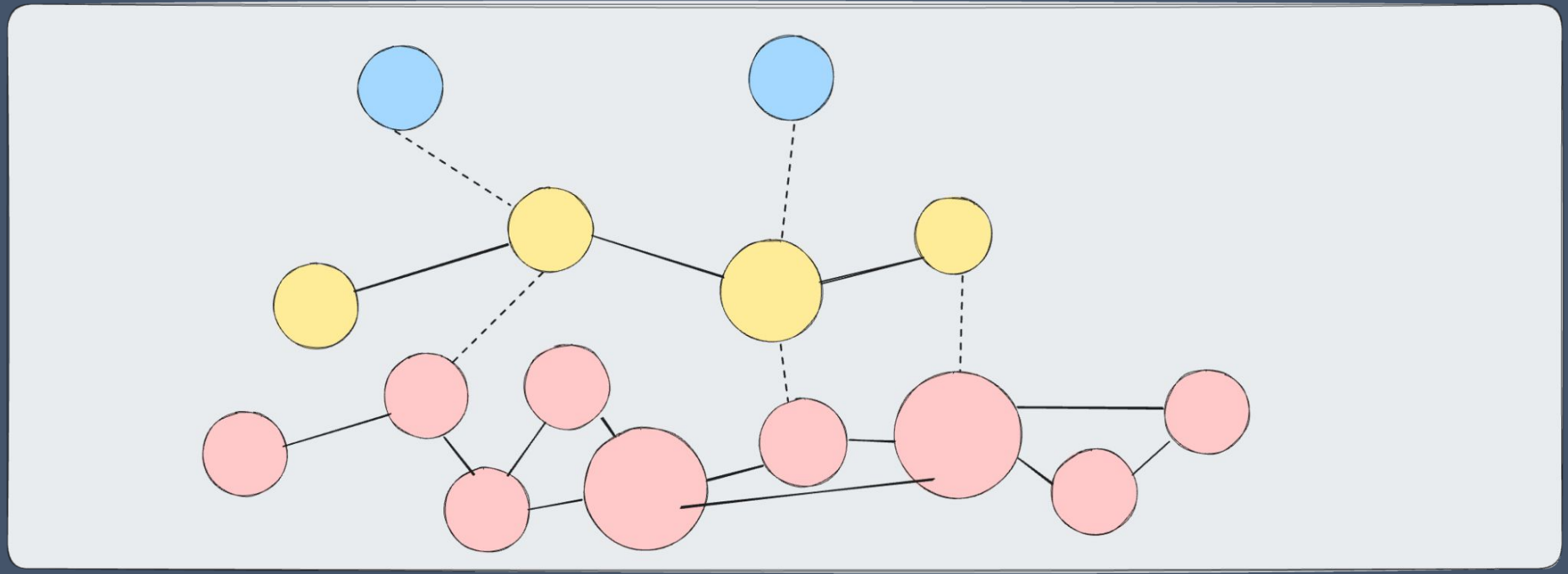
Inverted Indexes

```
"cat" → [doc1, doc3, doc7]
"dog" → [doc2, doc5]
"parrot" → [doc1, doc4]
```

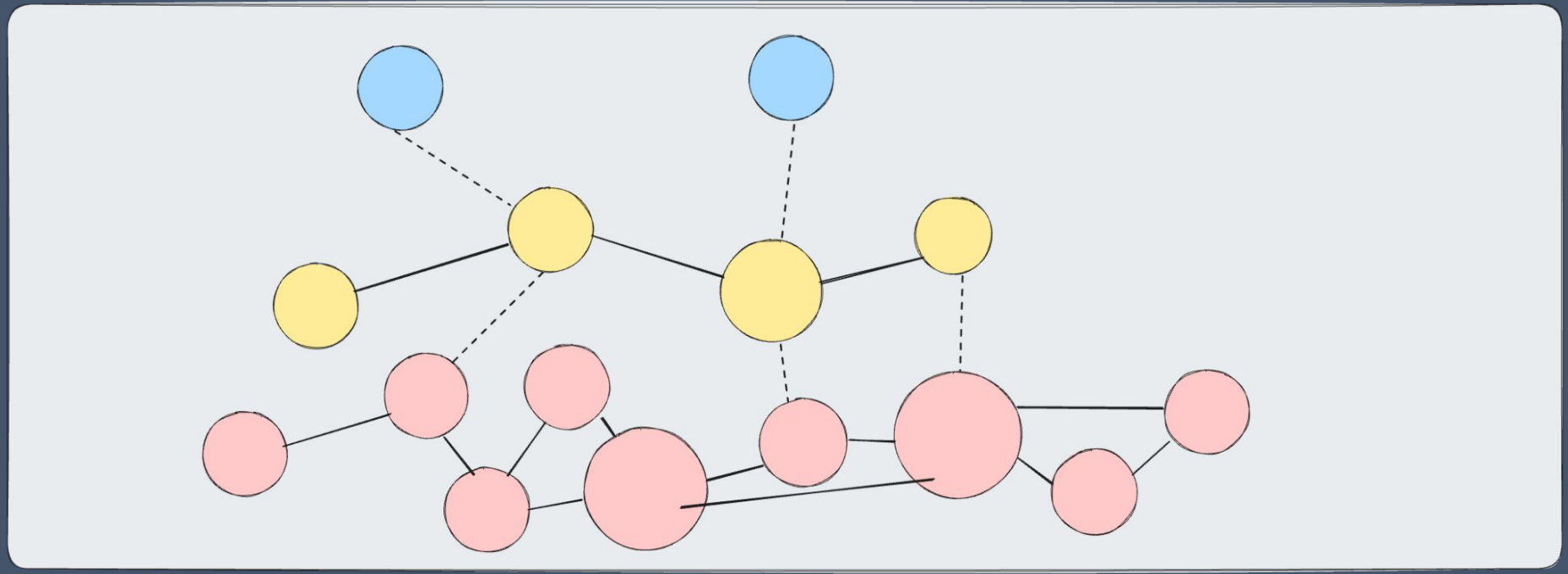


Approximate Nearest Neighbor (ANN)

A way to organize and filter vectors



Approximate Nearest Neighbor (ANN)
(Hierarchical Navigable Small World)



Bloom Filters - “so call me maybe”

No false negatives, but will result in mild overscans

```
"otel-collector-prod-01" → 31
"otel-collector-prod-10" → 31
```

Bloom Filters - “so call me maybe”

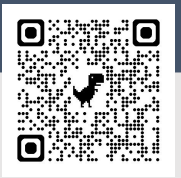
No false negatives, but will result in mild overscans

```
"otel-collector-prod-01" → 31
"otel-collector-prod-10" → 31
```

Much sparser than an inverted index - may fit entirely in memory

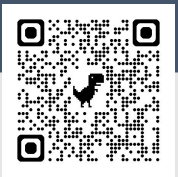
Prometheus (& friends)

Time-series Database



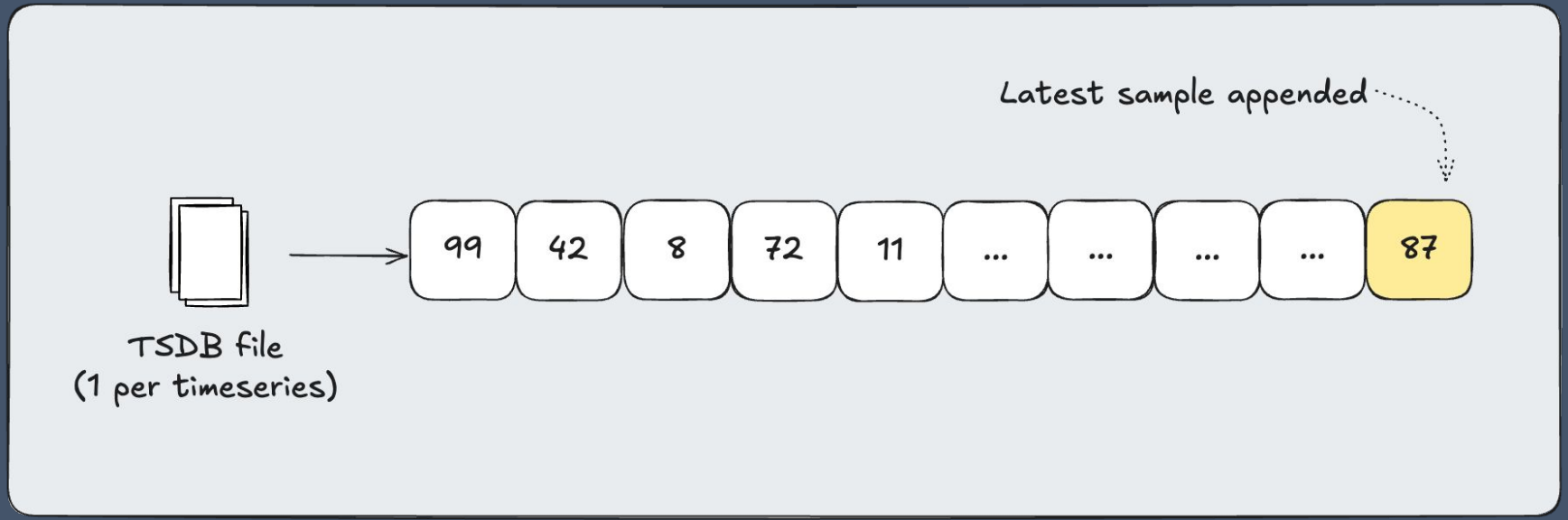
TSDB Blocks

Append-only



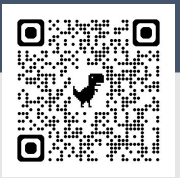
TSDB: Data is naturally ordered by time

Excellent for frequent reads of last-sample

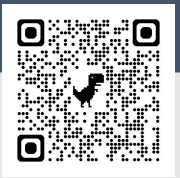


TSDB

No. of time series = cardinality^dimensionality

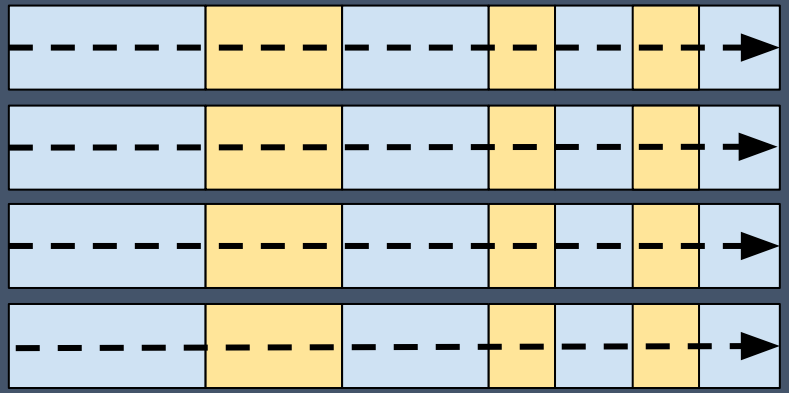


Row-oriented vs column-oriented storage



Row-oriented Storage

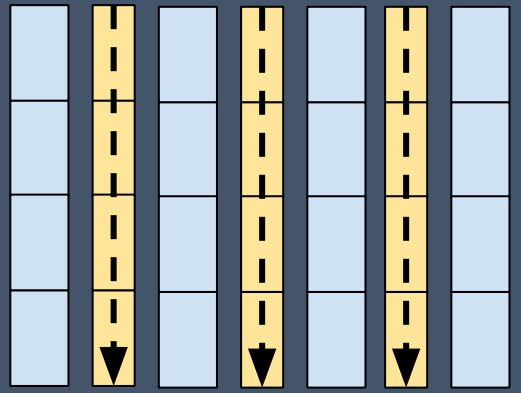
Read all columns in row



Rows compressed minimally or not at all

Column-oriented

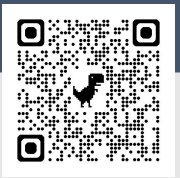
Read only selected columns

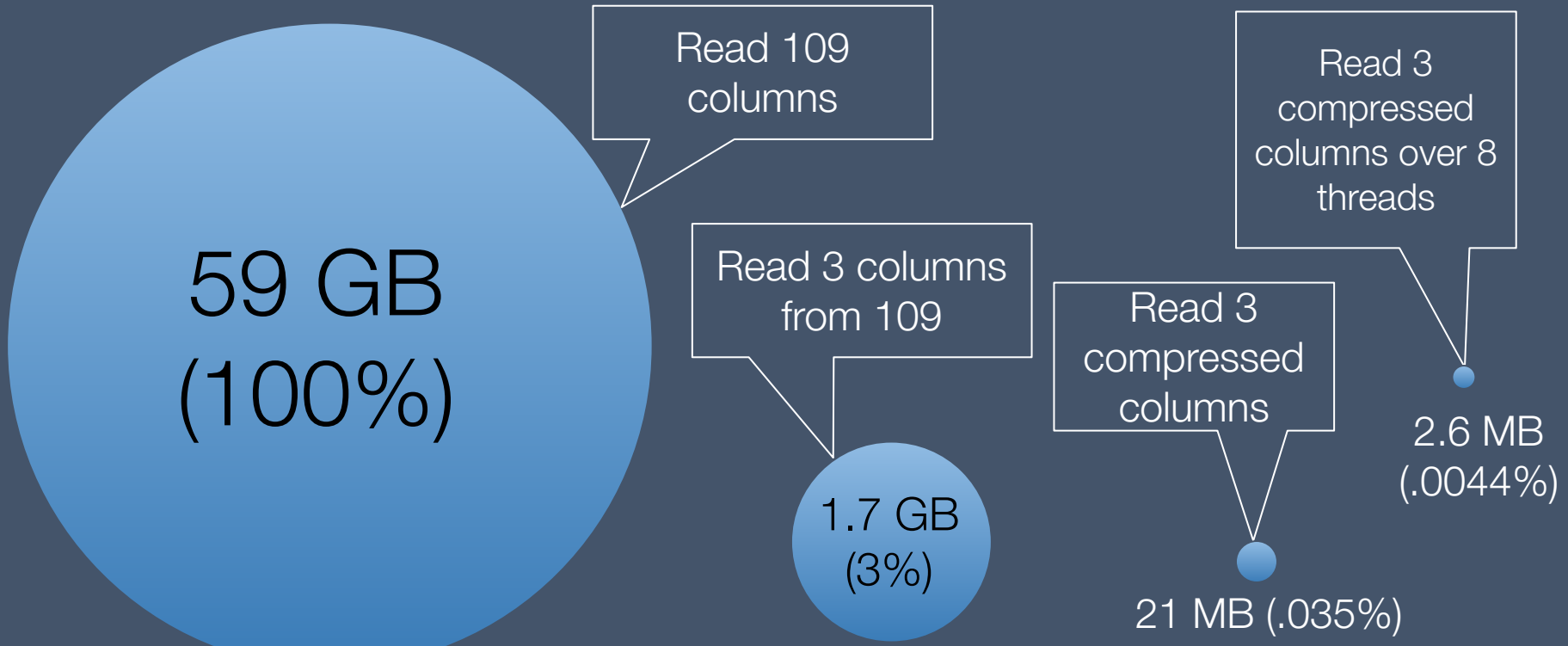


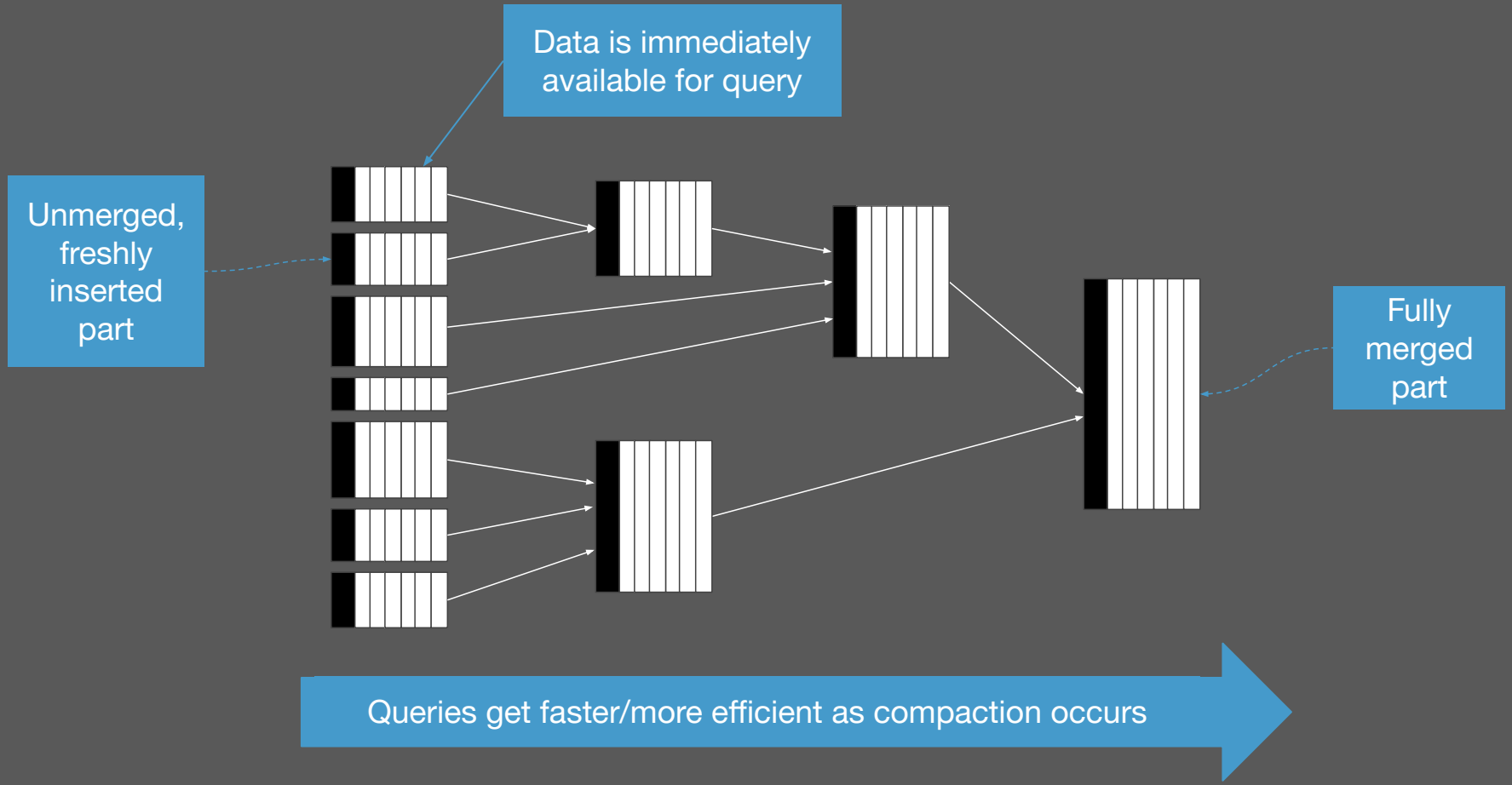
Columns highly compressed

ClickHouse

Column-oriented MergeTree

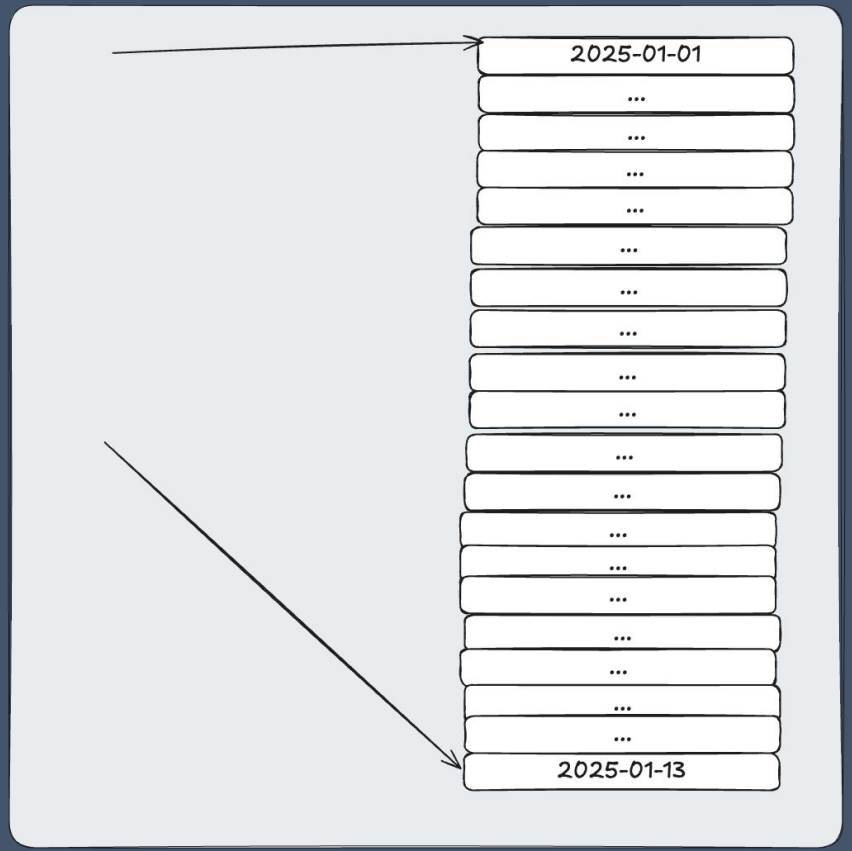



59 GB
(100%)
Read 109
columns
Read 3 columns
from 109
1.7 GB
(3%)
Read 3
compressed
columns
21 MB (.035%)
Read 3
compressed
columns over 8
threads
2.6 MB
(.0044%)

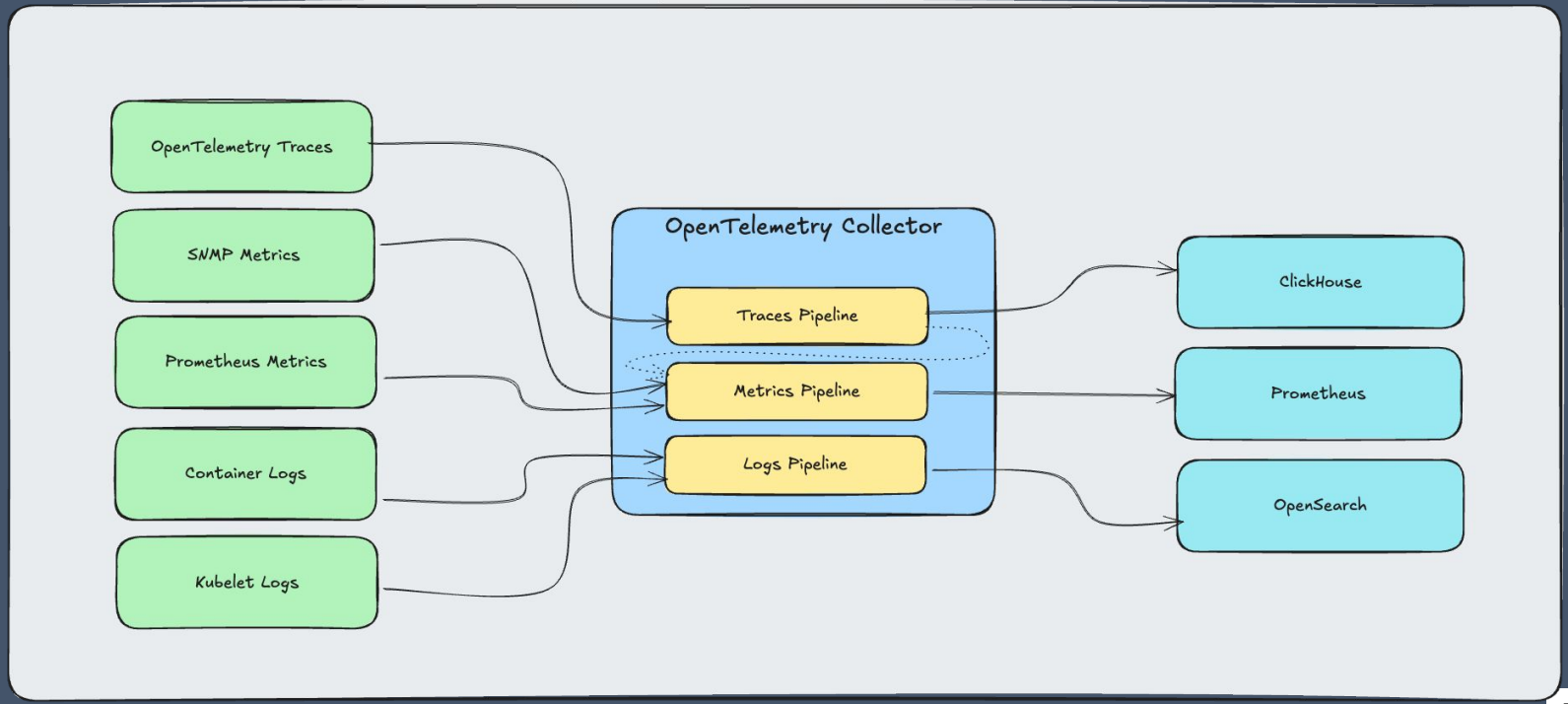

Data is immediately available for query
Unmerged, freshly inserted part
Fully merged part
Queries get faster/more efficient as compaction occurs

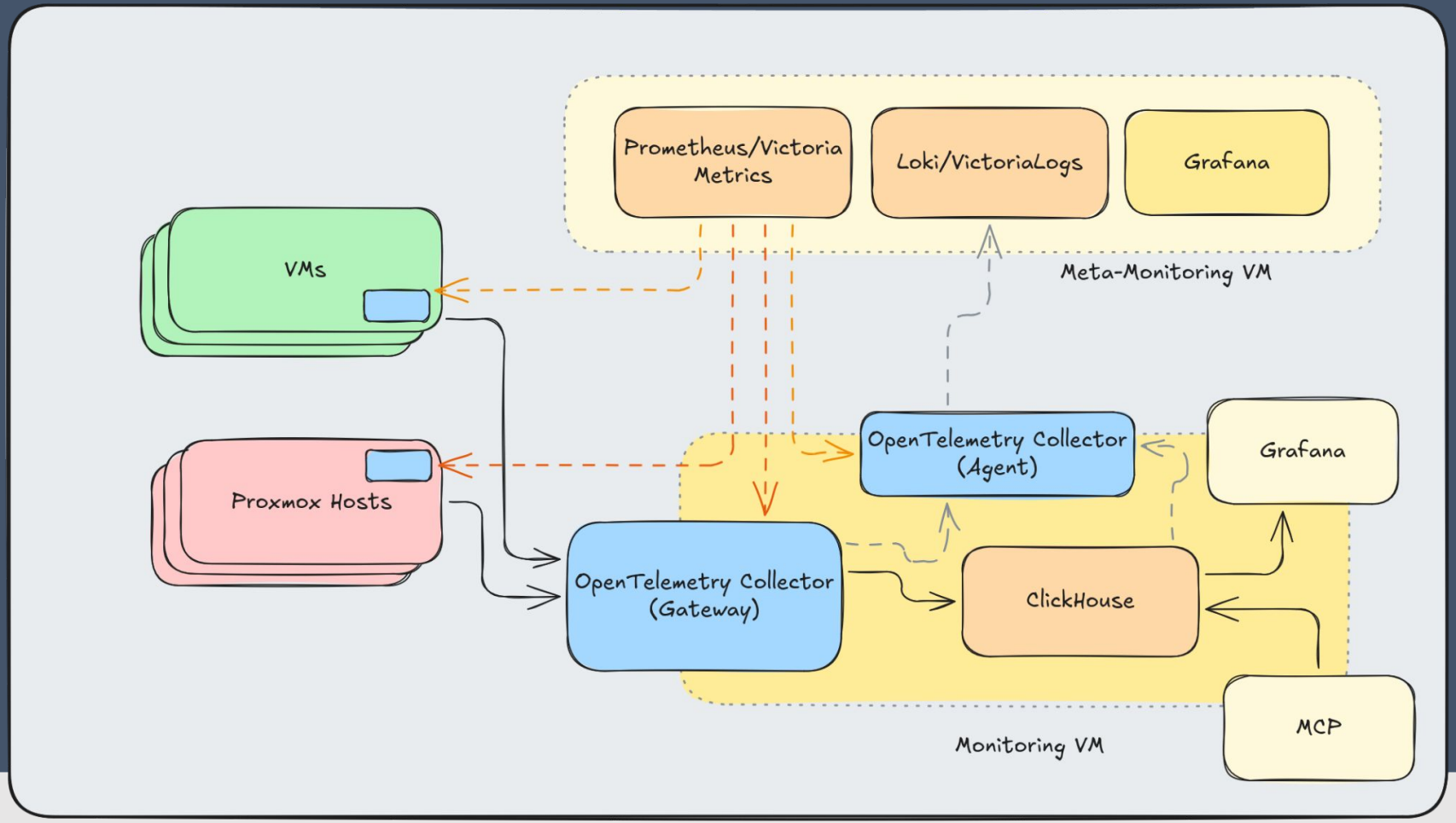
Sparse Indexes

Quickly & cheaply find data based on an ordered key



Which to choose?

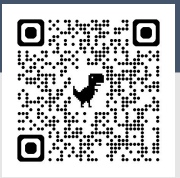



Prometheus/Victoria Metrics
Loki/VictoriaLogs
Grafana
Meta-Monitoring VM
VMs
Proxmox Hosts
OpenTelemetry Collector (Agent)
Grafana
OpenTelemetry Collector (Gateway)
ClickHouse
MCP
Monitoring VM

VictoriaMetrics

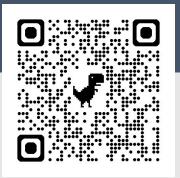
TSDB meets MergeTree

Josh Lee • https://joshuamlee.com/events/code-europe-26



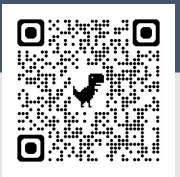
Loki

Uncomplicated logging (with label indexes)



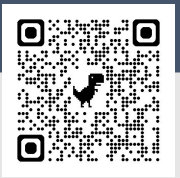
Honorable Mentions

Cortex, Thanos, Mimir, TimecaleDB, Solr, Druid...



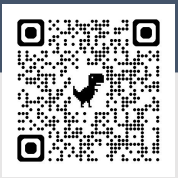
At (very) small scale

Just use what you have until it breaks (Postgres)



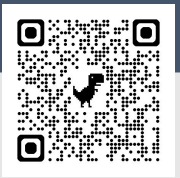
Hooked-on full-text search

Elastic/OpenSearch has your back



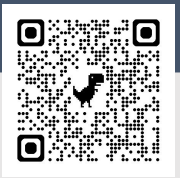
One database for everything

ClickHouse is pretty cool



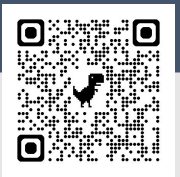
Wide-event analytics

ClickHouse is awesome

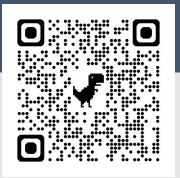


Lots of "last-sample" reads + alerts

Choose a TSDB like Prometheus or VictoriaMetrics



Database (Orientation)	**Style/QL**	**Storage**	**Indexes**	**Use Case**
Postgres (Row)	OLTP/SQL	Heap Pages	B-Tree	Update/upsert with guarantees
Cassandra (Columnar)	OLTP/CQL	Segments	Inverted, Bloom Filters, ANN	Scalable upserts
Prometheus (Columnar*)	TSDB/PromQL	TSDB files	By label	Time-series metrics, alerting
OpenSearch (Document)	Search/DSL	Lucene Segments	Inverted, ANN	Full-text search
ClickHouse (Columnar)	OLAP/SQL	MergeTree Parts	Sparse, Inverted, and more…	Real-time analytics



Happy querying!

Download slides, connect with me ➡️