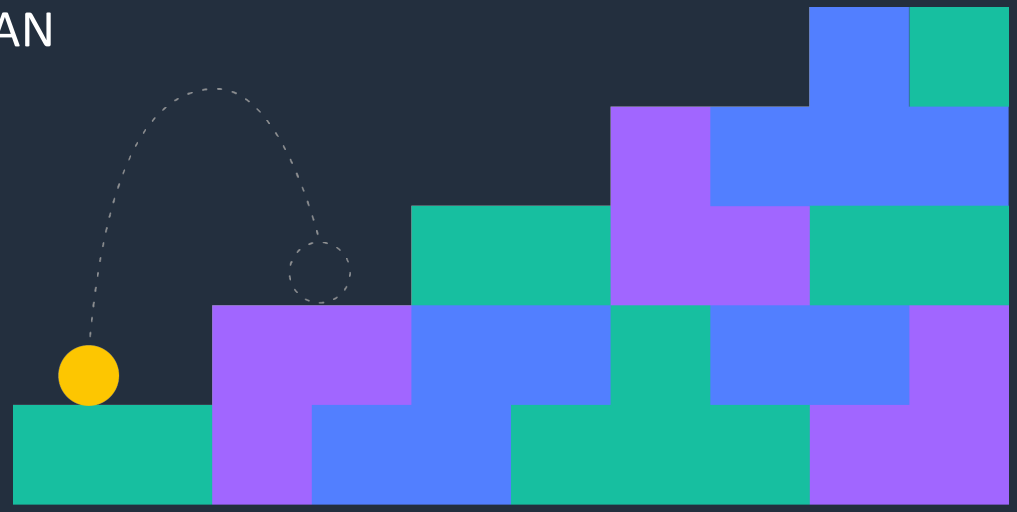
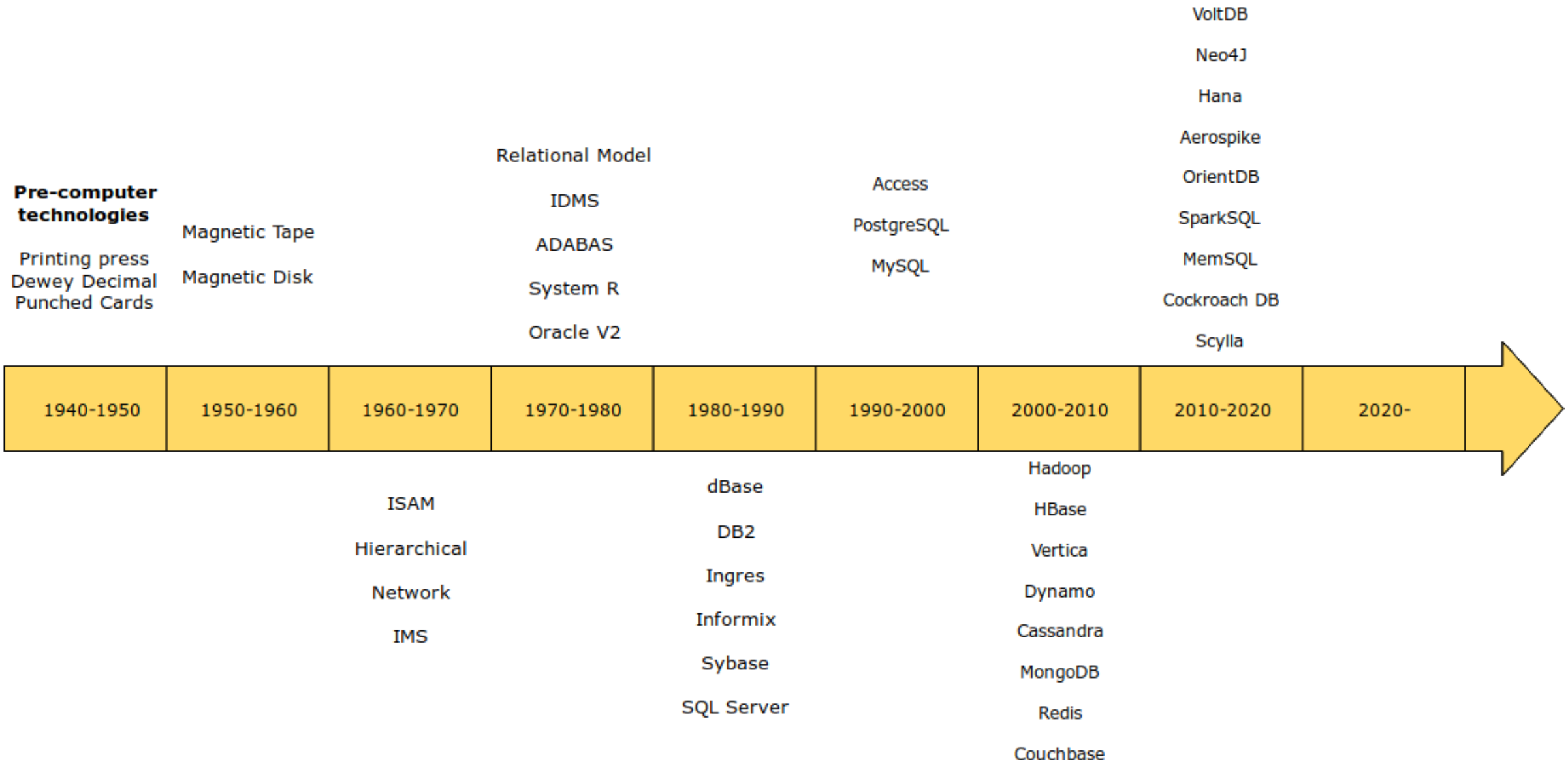
AWS Quick Start

AWS Purpose-Built Database Strategy: The Right Tool for The Right Job

Blair Layton, Business Development Manager, AWS, ASEAN

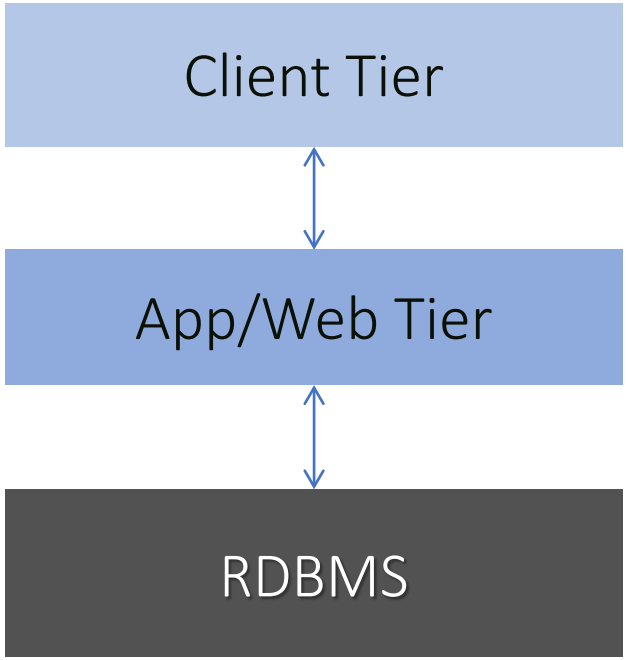


The Database Market is Changing



Traditional Database Architecture

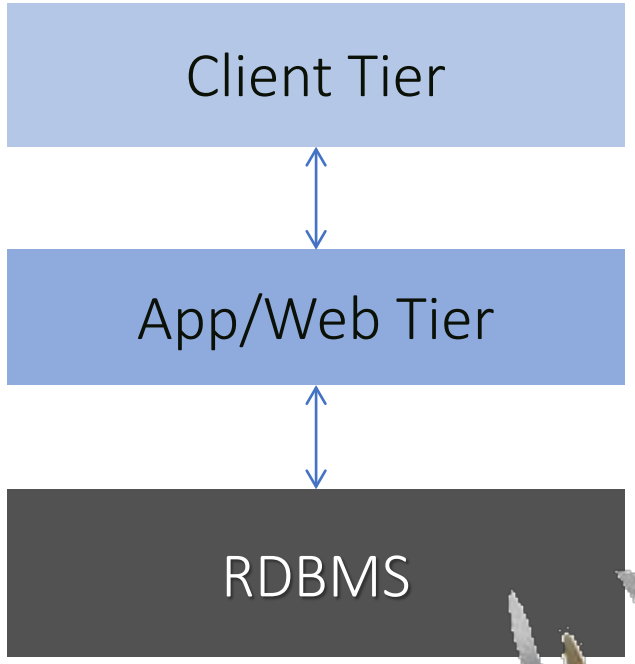
one database for all workloads

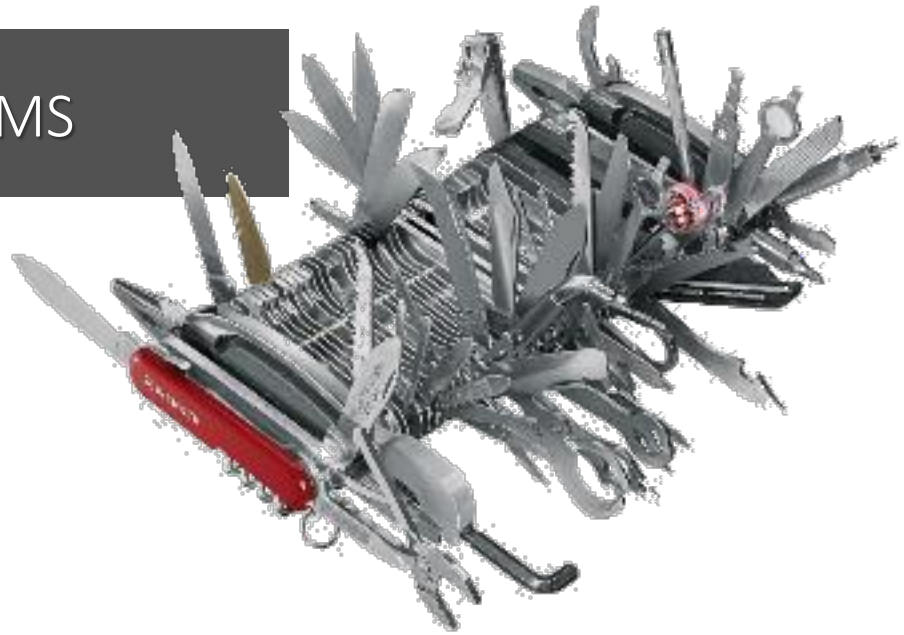


aws

Traditional Database Architecture

- key-value access
- complex queries
- transactions
- analytics





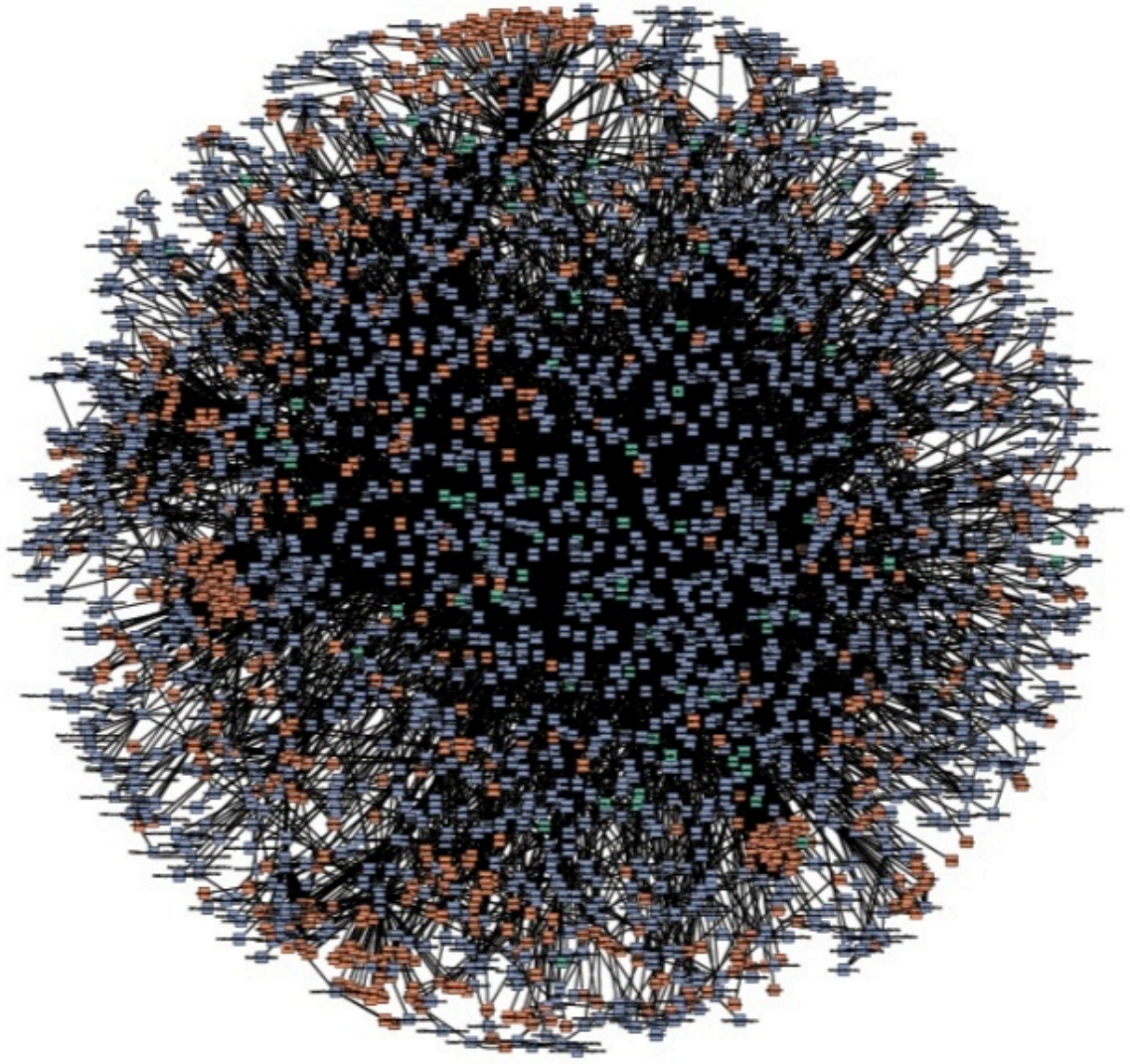
aws

Data categories and common use cases

 Relational	 Key-value	 Document	 In-memory	 Graph	 Search	 Time-series	 Ledger
Referential integrity, ACID transactions, schema-on-write	Low-latency, key lookups with high throughput and fast ingestion of data	Indexing and storing documents with support for query on any attribute	Microseconds latency, key-based queries, and specialized data structures	Creating and navigating data relations easily and quickly	Indexing and searching semi-structured logs and data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data
Lift and shift, EMR, CRM, finance	Real-time bidding, shopping cart, social	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	Product catalog, help and FAQs, full text	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial

Application Architecture is Changing

Microservices at Amazon



Service-Oriented Architecture (SOA)

Single-purpose

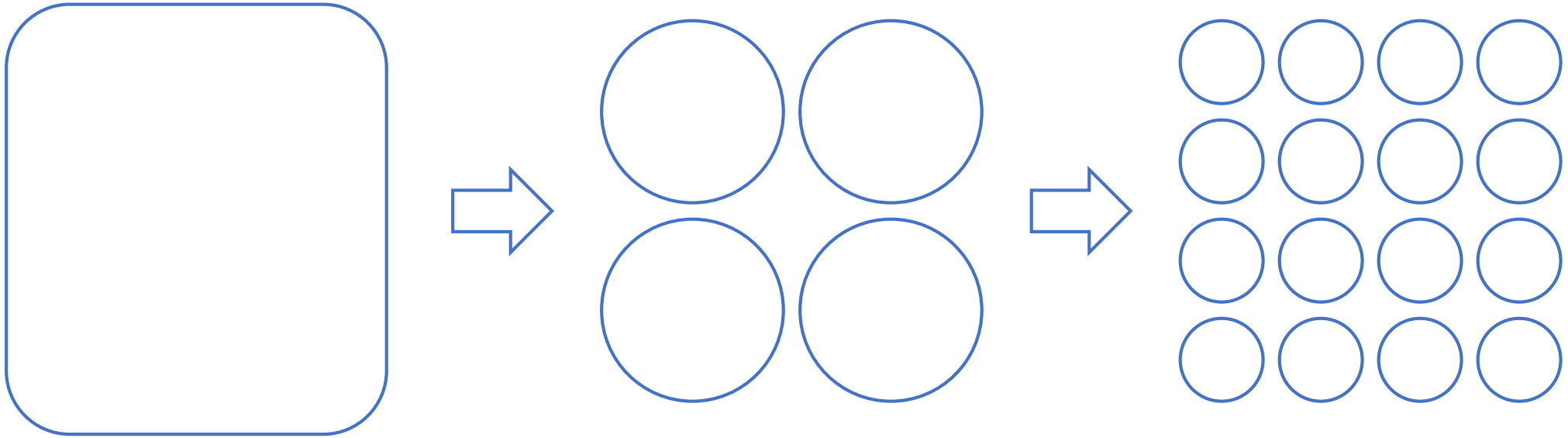
Connect only through APIs

Connect over HTTPS

“Microservices”

aws

Monolithic vs. SOA vs. Microservices



Monolithic
Single Unit

SOA
Coarse-grained

Microservices
Fine-grained

aws

Microservices…

Eliminates any long-term commitment to a technology stack

Polyglot ecosystem

Polyglot persistence

- Decompose Databases
- Database per microservice pattern

Allows easy use of Canary and Blue-Green deployments

Purpose-Built Databases

Our portfolio

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning

- QuickSight
- SageMaker

Analytics

- Redshift — Data warehousing
- EMR — Hadoop + Spark
- Athena — Interactive analytics
- Elasticsearch Service — Operational Analytics
- Kinesis Data Analytics — Real time

Databases

- Aurora — MySQL, PostgreSQL
- DynamoDB — Key value, Document
- QLDB — Ledger Database — NEW
- RDS — MySQL, PostgreSQL, MariaDB, Oracle, SQL Server
- ElastiCache — Redis, Memcached
- Timestream — Time Series — NEW
- RDS on VMware — NEW
- Neptune — Graph
- DocumentDB — Document — NEW

Blockchain

- Managed Blockchain — NEW
- Blockchain Templates

Data Lake

- S3/Glacier
- Lake Formation — Data Lakes — NEW
- Glue — ETL & Data Catalog

Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

AWS: Purpose-built databases

Relational	Key-value	Document	In-memory	Graph	Search	Time-series	Ledger
Amazon RDS	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Elasticsearch Service	Amazon Timestream	Amazon Quantum Ledger Database

Amazon RDS: Aurora (MySQL, PostgreSQL) · Community (MySQL, PostgreSQL, MariaDB) · Commercial (Oracle, Microsoft SQL Server)

Amazon ElastiCache: redis · memcached

Relational Databases

Amazon RDS

Managed relational database service with a choice of popular database engines

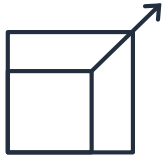
Amazon Aurora | MySQL | PostgreSQL | MariaDB | Microsoft SQL Server | ORACLE

Easy to administer

No need to provision infrastructure, install, and maintain DB software

Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, and failover



Highly scalable

Scale DB compute and storage with a few clicks; minimal downtime for your application

Fast & secure

SSD storage and guaranteed provisioned I/O; data encryption at rest and in transit

aws

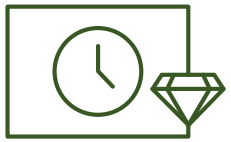
Amazon Aurora

MySQL and PostgreSQL compatible relational database built for the cloud

Performance and availability of commercial-grade databases at 1/10th the cost

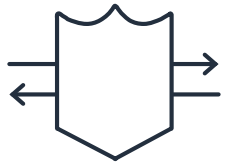
Performance & scalability

5x throughput of standard MySQL and 3x of standard PostgreSQL; scale-out up to 15 read replicas



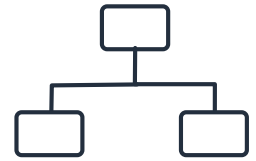
Availability & durability

Fault-tolerant, self-healing storage; six copies of data across three AZs; continuous backup to S3



Highly secure

Network isolation, encryption at rest/transit



Fully managed

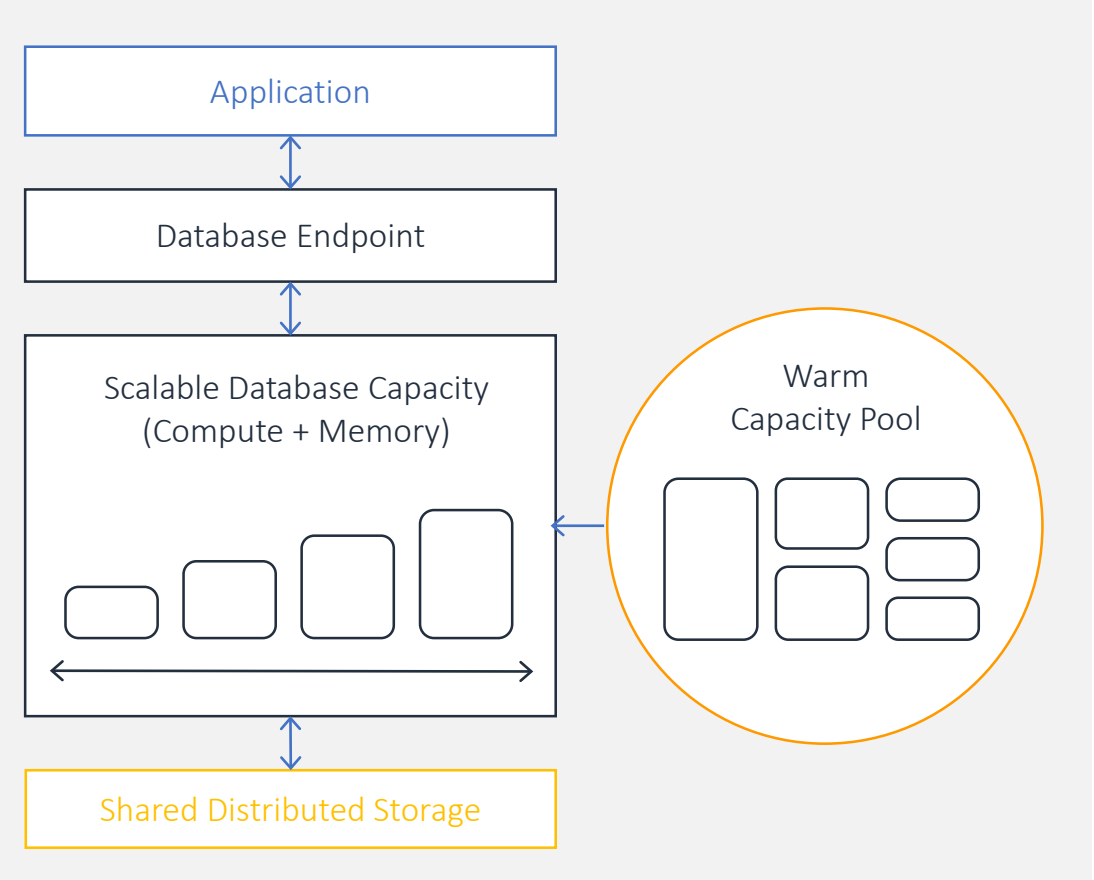
Managed by RDS: no hardware provisioning, software patching, setup, configuration, or backups

Aurora: Fastest Growing Service in AWS History

aws

Aurora Serverless

On-demand, auto-scaling database for applications with variable workloads



Starts up on demand, shuts down when not in use

Automatically scales with no instances to manage

Pay per second for the database capacity you use

Non-Relational Databases

Amazon DynamoDB

We needed to adapt to power Amazon.com

Dynamo: Amazon's Highly Available Key-value Store

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels

Amazon.com

ORIGINAL

ABSTRACT

Reliability at massive scale is one of the biggest challenges we face at Amazon.com, one of the largest e-commerce operations in the world; even the slightest outage has significant financial consequences and impacts customer trust. The Amazon.com platform, which provides services for many web sites worldwide, is implemented on top of an infrastructure of tens of thousands of servers and network components located in many datacenters around the world. At this scale, small and large components fail continuously and the way persistent state is managed in the face of these failures drives the reliability and scalability of the software systems.

This paper presents the design and implementation of Dynamo, a highly available key-value storage system that some of Amazon's core services use to provide an "always-on" experience. To achieve this level of availability, Dynamo sacrifices consistency under certain failure scenarios. It makes extensive use of object versioning and application-assisted conflict resolution in a manner that provides a novel interface for developers to use.

One of the lessons our organization has learned from operating Amazon's platform is that the reliability and scalability of a system is dependent on how its application state is managed. Amazon uses a highly decentralized, loosely coupled, service oriented architecture consisting of hundreds of services. In this environment there is a particular need for storage technologies that are always available. For example, customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados. Therefore, the service responsible for managing shopping carts requires that it can always write to and read from its data store, and that its data needs to be available across multiple data centers.

Dealing with failures in an infrastructure comprised of millions of components is our standard mode of operation; there are always a small but significant number of server and network components that are failing at any given time. As such Amazon's software systems need to be constructed in a manner that treats failure handling as the normal case without impacting availability or performance.

Needed to power Amazon.com

Required massive scalability and reliability

DynamoDB designed to meet this need

aws

Many Applications Require Millisecond Latency at Any Scale

Example: Amazon Prime Day 2017

Biggest shopping event in Amazon history

Thousands of Amazon teams using DynamoDB

3.34 trillion requests

Peaked at **12.9 million** requests per second

aws

Amazon DynamoDB

Fully-managed nonrelational database for any scale

Fully managed

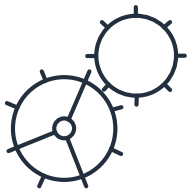
Maintenance-free

Serverless

Auto scaling

Backup and restore

Global tables

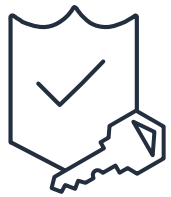


High performance

Fast, consistent performance

Virtually unlimited throughput

Virtually unlimited storage



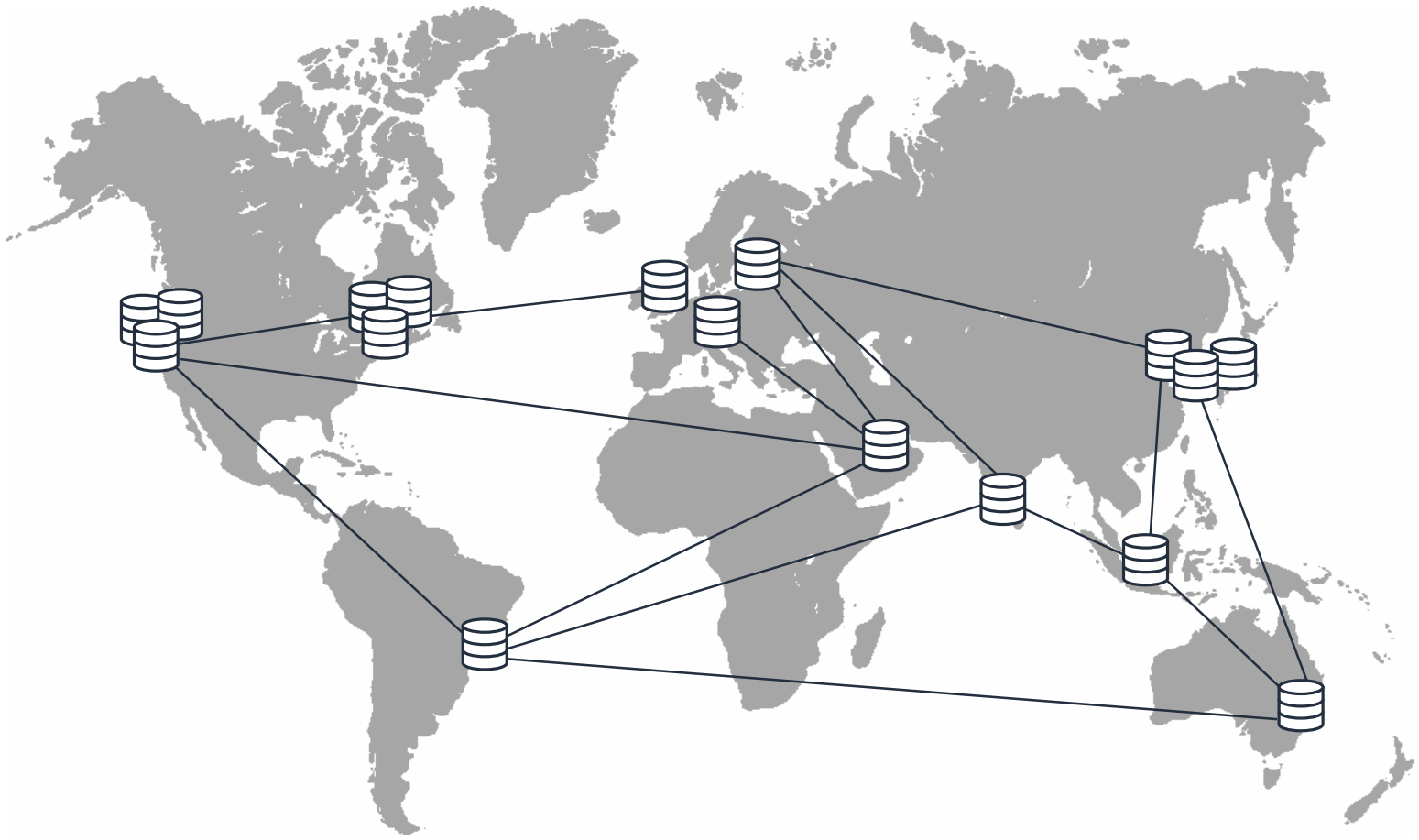
Secure

Encryption at rest and transit

Fine-grained access control

PCI, HIPAA, FIPS140-2 eligible

DynamoDB Global Tables



Build high-performance, globally distributed applications

Low latency reads and writes to locally available tables

Multi-region redundancy and resiliency

Easy to set up and no application rewrites required

Amazon ElastiCache

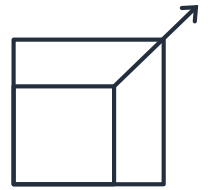
Fully-managed, Redis or Memcached compatible, low-latency, in-memory data store

Extreme performance

In-memory data store and cache
for sub-millisecond response times

Fully managed

AWS manages all hardware
and software setup,
configuration, monitoring



Easily scalable

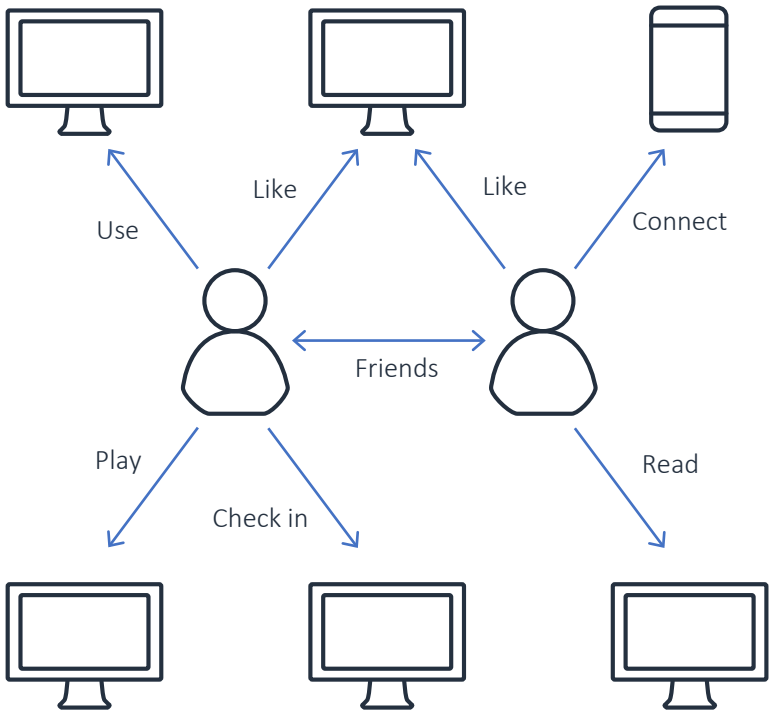
Read scaling with replicas

Write and memory scaling
with sharding

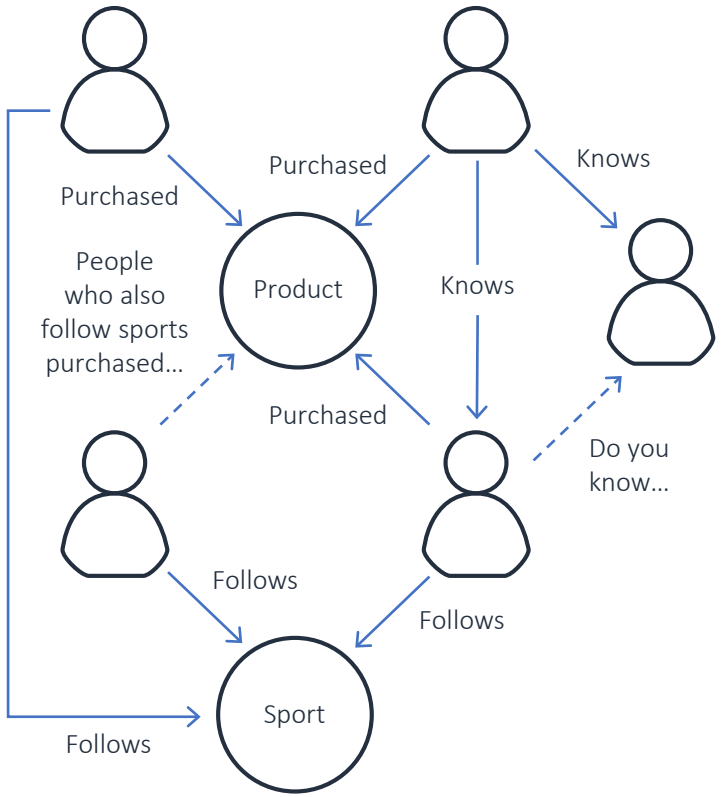
Non-disruptive scaling

Graph Use Cases

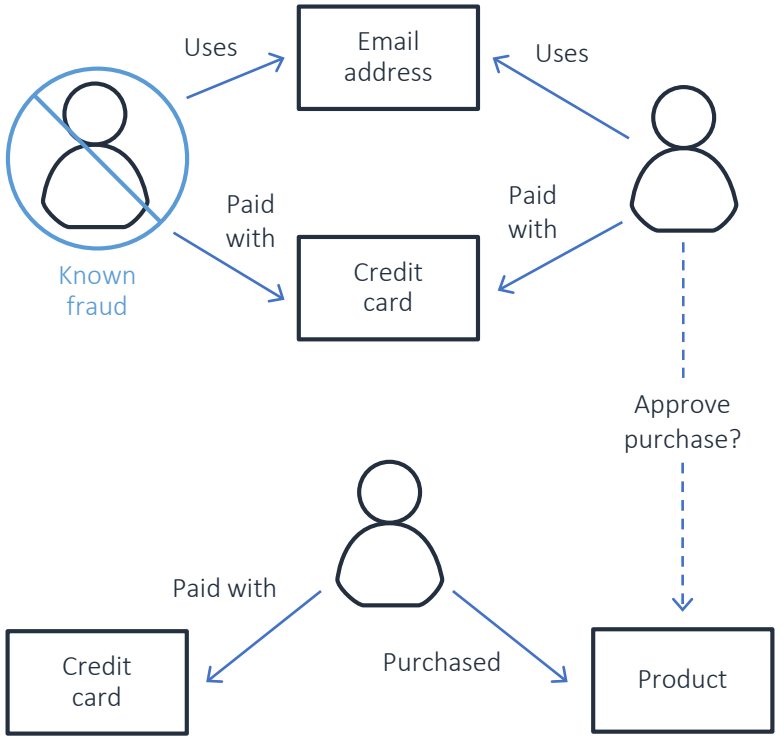
Social news feed



Recommendations



Retail fraud detection

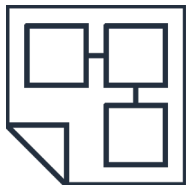


Challenges Building Apps with Highly-Connected Data

Relational databases

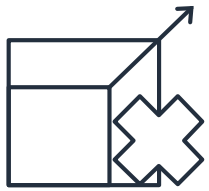
Unnatural for querying graph

Inefficient graph processing

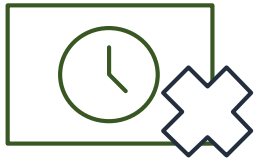


Rigid schema inflexible for changing graphs

Existing graph databases

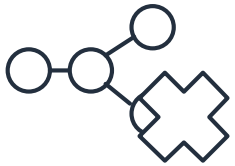


Difficult to scale



Difficult to maintain high availability

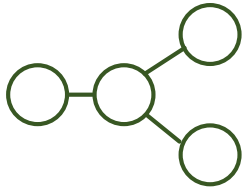
Too expensive



Limited support for open standards

Amazon Neptune

Fully managed graph database

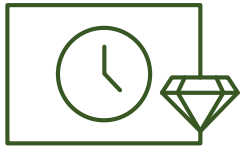


Open

Supports Apache TinkerPop and W3C RDF graph models

Fast & scalable

Store billions of relationships; query with millisecond latency



Reliable

Six replicas of your data across three AZs with full backup and restore

Easy

Build powerful queries easily with Gremlin and SPARQL

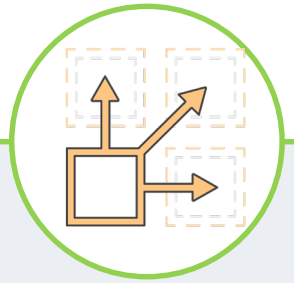
Amazon DocumentDB

Fast, scalable, and fully managed MongoDB-compatible database service

Fast

Millions of requests per second with millisecond latency; twice the throughput of MongoDB

Scalable



Separation of compute and storage enables both layers to scale independently; scale out to 15 read replicas in minutes

Fully managed

Managed by AWS: no hardware provisioning; auto patching, quick setup, secure, and automatic backups

MongoDB compatible

Compatible with MongoDB 3.6; use the same SDKs, tools, and applications with Amazon DocumentDB

Database Migration Service

AWS Database Migration Service

Migrating Databases to AWS

100,000+
Databases migrated

Migrate between on-premises and AWS

Migrate between databases

Automated schema conversion

Data replication for zero-downtime migration

100,000+ Databases Migrated with DMS

aws

Thank You for Attending AWS Quick Start

We hope you found it interesting! A kind reminder to **complete the survey.**
Let us know what you thought of today's event and how we can improve the event experience for you in the future.


aws-apac-marketing@amazon.com


twitter.com/AWSCloud


facebook.com/AmazonWebServices


youtube.com/user/AmazonWebServices


slideshare.net/AmazonWebServices


twitch.tv/aws