Cloud Trends

The evolution of culture and technology

Adrian Cockcroft @adrianco
VP Cloud Architecture Strategy
Amazon Web Services

amazon web services

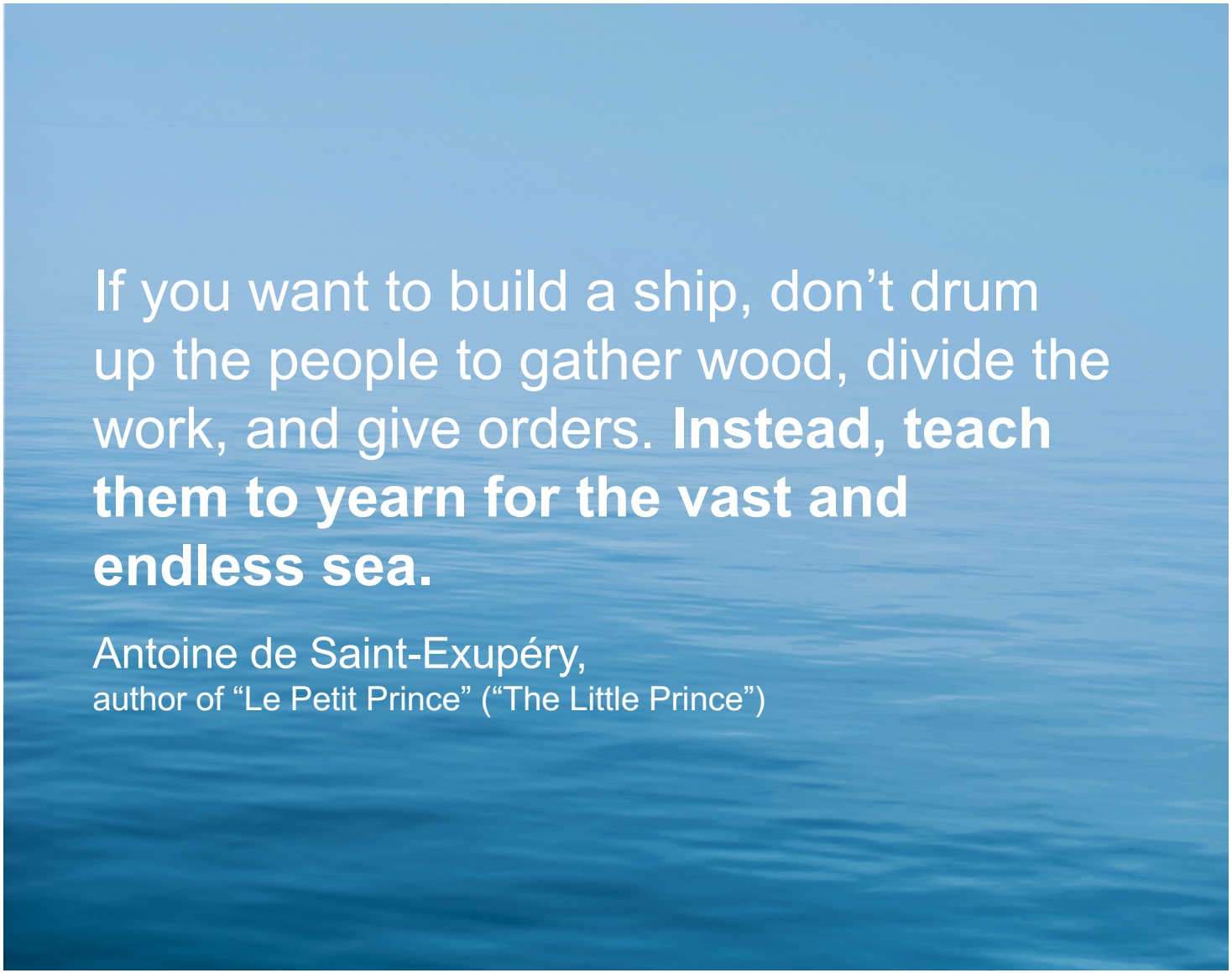
Culture
And Evolution

1. Get the culture right
2. Migrate to the cloud
3. The new de-normal, untangle data tier
4. Monoliths to microservices to functions

Cloud Trends

1. Get the culture right
2. Migrate to the cloud
3. The new de-normal, untangle data tier
4. Monoliths to microservices to functions
5. Open source and artificial intelligence

Culture


If you want to build a ship, don’t drum up the people to gather wood, divide the work, and give orders. **Instead, teach them to yearn for the vast and endless sea.**
Antoine de Saint-Exupéry,
author of “Le Petit Prince” (“The Little Prince”)

NORDSTROM

Culture

NETFLIX

Culture

Seven Aspects
of Netflix Culture

1. Values are what we value
2. High performance
3. Freedom & responsibility
4. Context, not control
5. Highly aligned, loosely coupled
6. Pay top of market
7. Promotions & development

amazon

Culture

Amazon leadership principles

- Customer obsession
- Ownership
- Invent and simplify
- Are right, a lot
- Hire and develop the best
- Insist on the highest standards
- Think big
- Bias for action
- Frugality
- Learn and be curious
- Earn trust of others
- Dive deep
- Have backbone; disagree and commit
- Deliver results

Culture

Intentional

Appropriate

Judgement

Migrating to Cloud

Lessons from the Netflix cloud journey, brought up to date

2008
Start with a shock

IT’s assumption: make systems perfect
so that developers don't have to think about failures

High-end IBM P-series hardware, Oracle...

Two-day outage caused by SAN hardware failure!

2008

Question

Assumption

Failure raised questions…

Availability has to be application concern!

Use low cost cloud infrastructure?

2009
Add an existential threat

Vast increase in datacenter capacity was needed

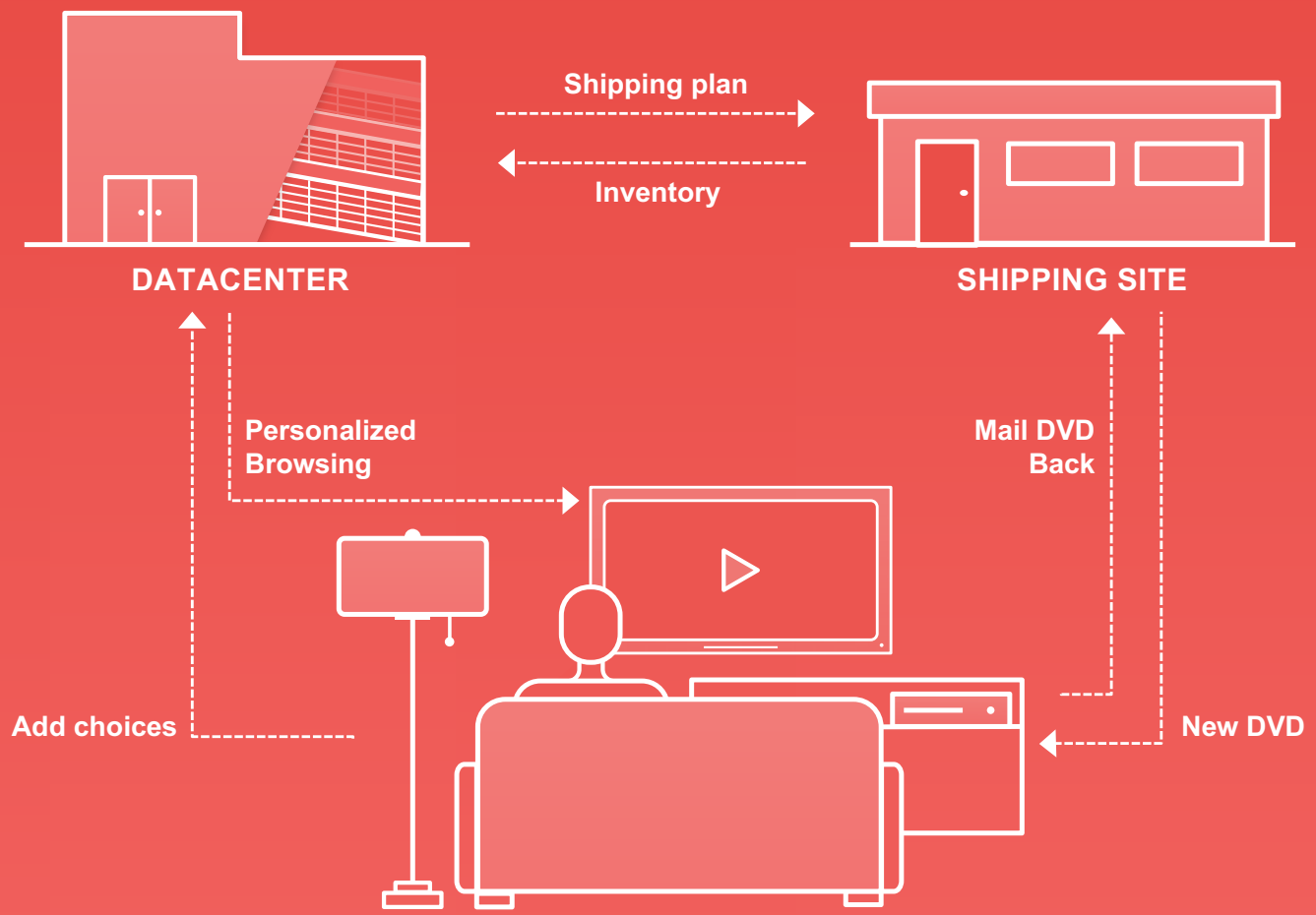
Unpredictable in advance, how much, where…

Why? Online streaming replacing DVD shipping logistics

Nowadays, systems of engagement are dominating IT

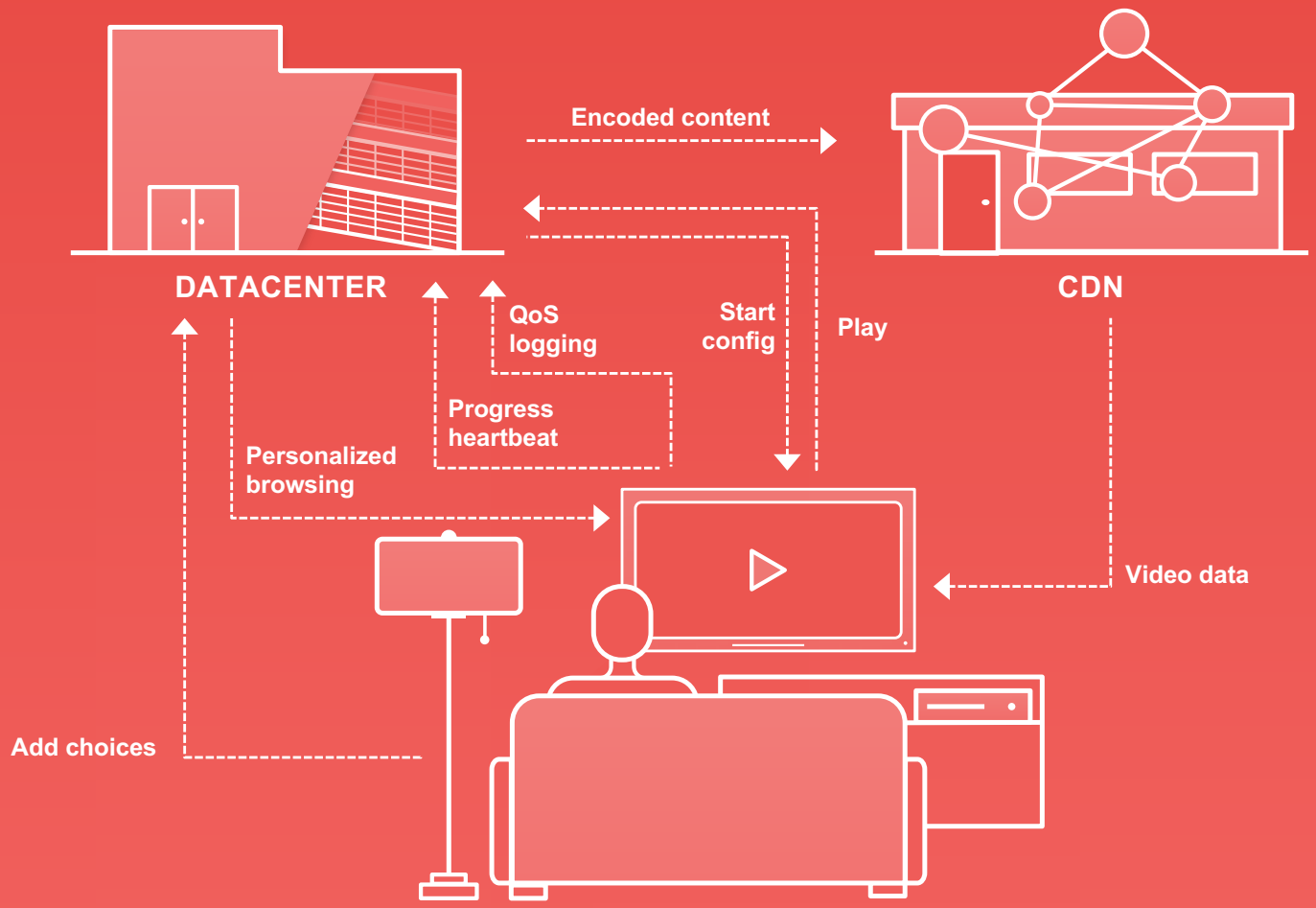
DVD Business

A few interactions per week per customer to datacenter



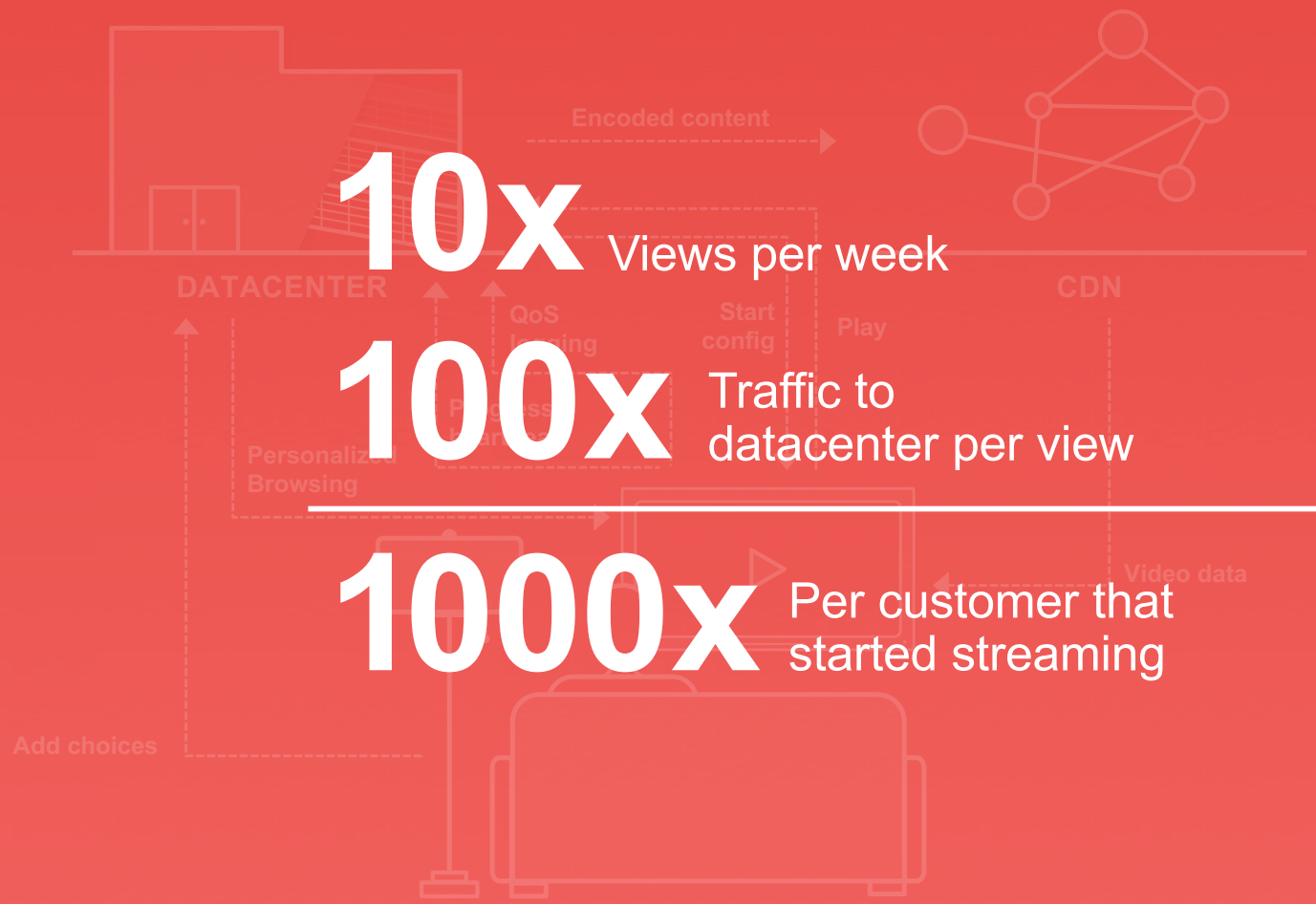
DVD ıming Business

Binge watching episodes of TV shows every day



Streaming Business

Binge watching episodes of TV shows every day

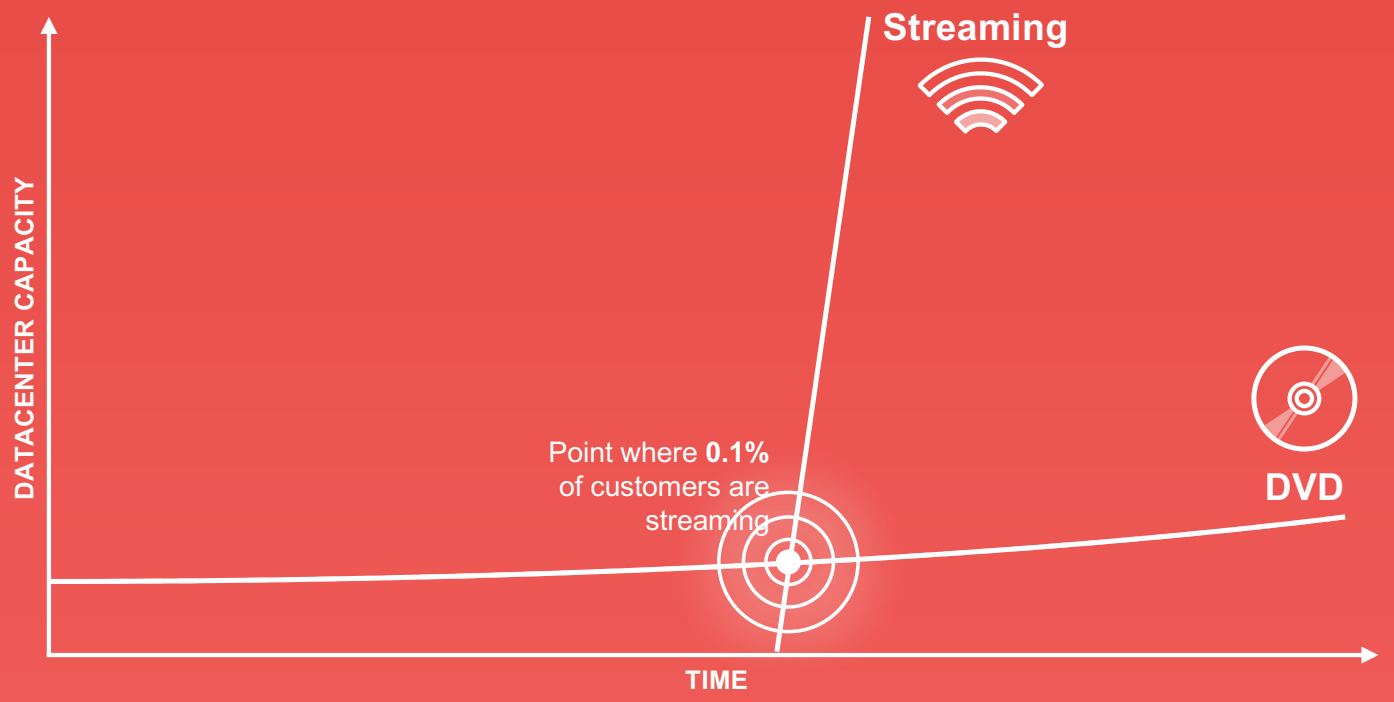


10x Views per week

100x Traffic to datacenter per view

1000x Per customer that started streaming

The Digital Transformation Capacity Crunch



If we say new workload causes 1000x traffic to datacenter, then when 0.1% of users switch, the capacity needed is equal.

Choices

Recruit world class datacenter operations build team and guess how much capacity they would need, and build it before it was needed — lots of upfront $$$ spend

OR

Use the Elastic Compute service of AWS, built by one of Netflix biggest competitors, and spend $$$ on video content and developers

2009

Mitigate risks

Competition
Understand how AWS was separated from Amazon Prime

Capacity
Experiments to see what worked

Business
Sign Up For Enterprise License Agreement

Publicity
NYT story about Netflix and AWS April 2010

The New York Times

TECHNOLOGY

Companies Slowly Join Cloud-Computing

By BRAD STONE and ASHLEE VANCE APRIL 18, 2010

SAN FRANCISCO — This year, Netflix made what looked like a peculiar choice: the DVD-by-mail company decided that over the next two years, it would move most of its Web technology — customer movie queues, search tools and the like — over to the computer servers of one of its chief rivals, Amazon.com.

Amazon, like Netflix, wants to deliver movies to people's homes over the Internet. But the online retailer, based in Seattle, has lately gained traction with a considerably more ambitious effort: the business of renting other companies the remote use of its technology infrastructure so they can run their computer operations. In the parlance of technophiles, they would operate "in the cloud."

2009

Applications

Encoding movies
Big backlog, not enough capacity

Moved to AWS EC2
Showed that capacity existed on demand

Shut down capacity to save as backlog varied

2009
Applications

Quality of Service (QoS) logging
Too much traffic to datacenter databases

Storage for logs moved to S3
Unlimited space

Log analysis moved to EMR - Hadoop
Worked with AWS to support Hadoop + Hive
in Elastic Map-Reduce service

Crunch Time

Start of 2010
Decided not to build any more datacenter capacity

Need to move to AWS before end of 2010 to survive

AWS

Front end on AWS

Web pages and API clients migrate to cloud

Backend capacity expands to fill remaining space

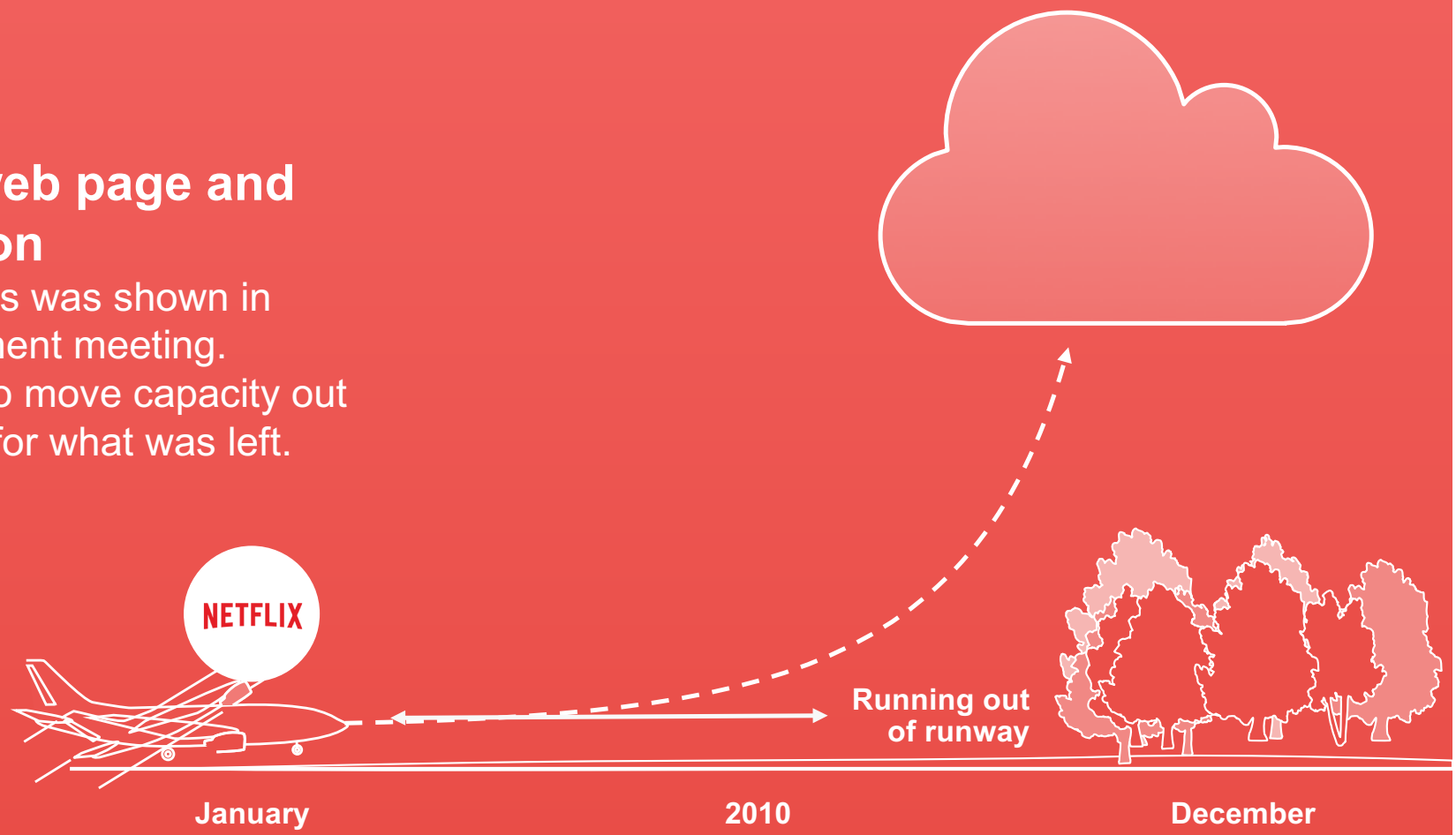
Most backend still in datacenter

DATACENTER

January **2010** **December**

Front end web page and API migration

A picture like this was shown in every management meeting.
Hard deadline to move capacity out to make space for what was left.



Migration Sequence

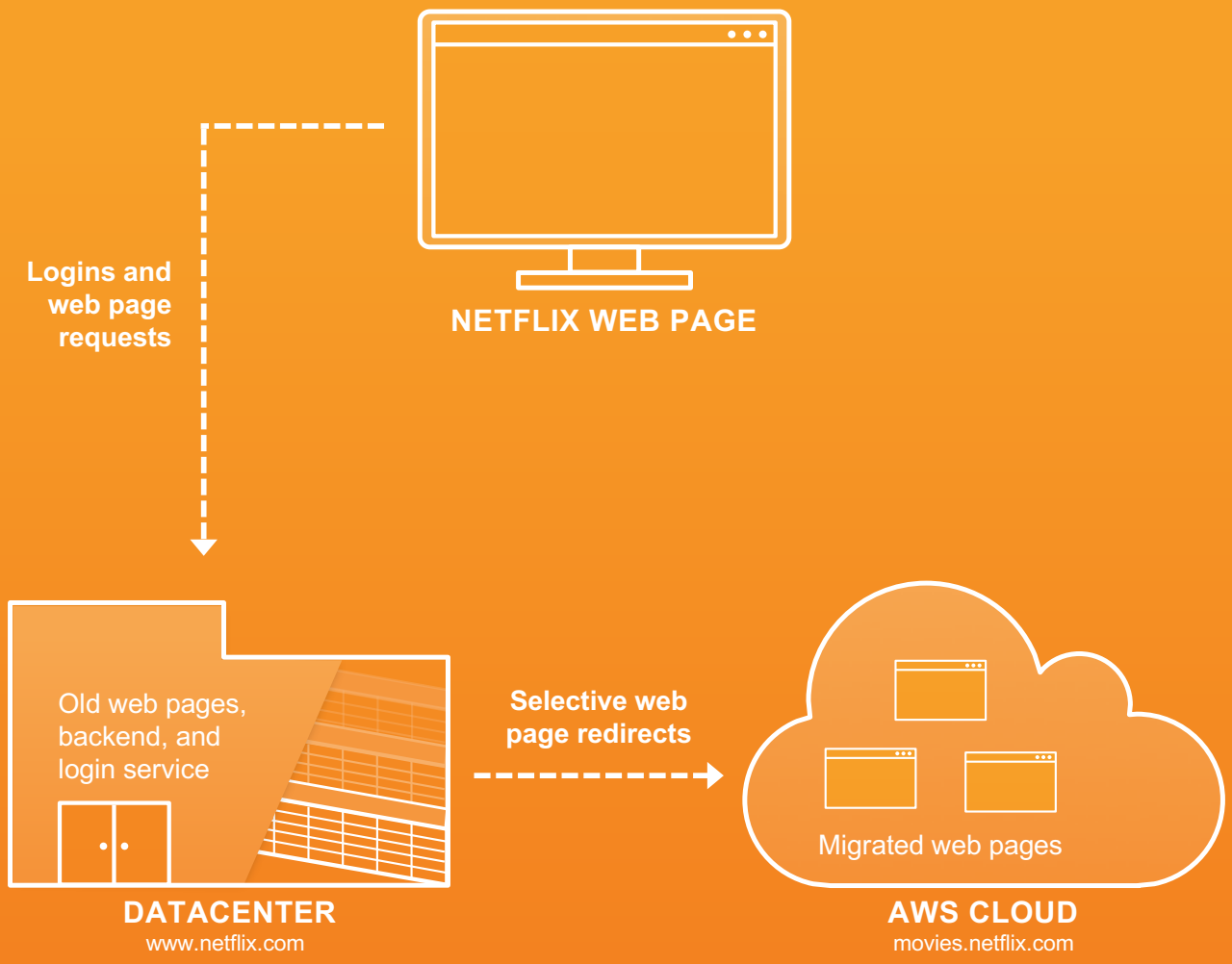
Start with the simplest possible API service

Next the simplest web page

Then pages and APIs one by one

How to Run Both?

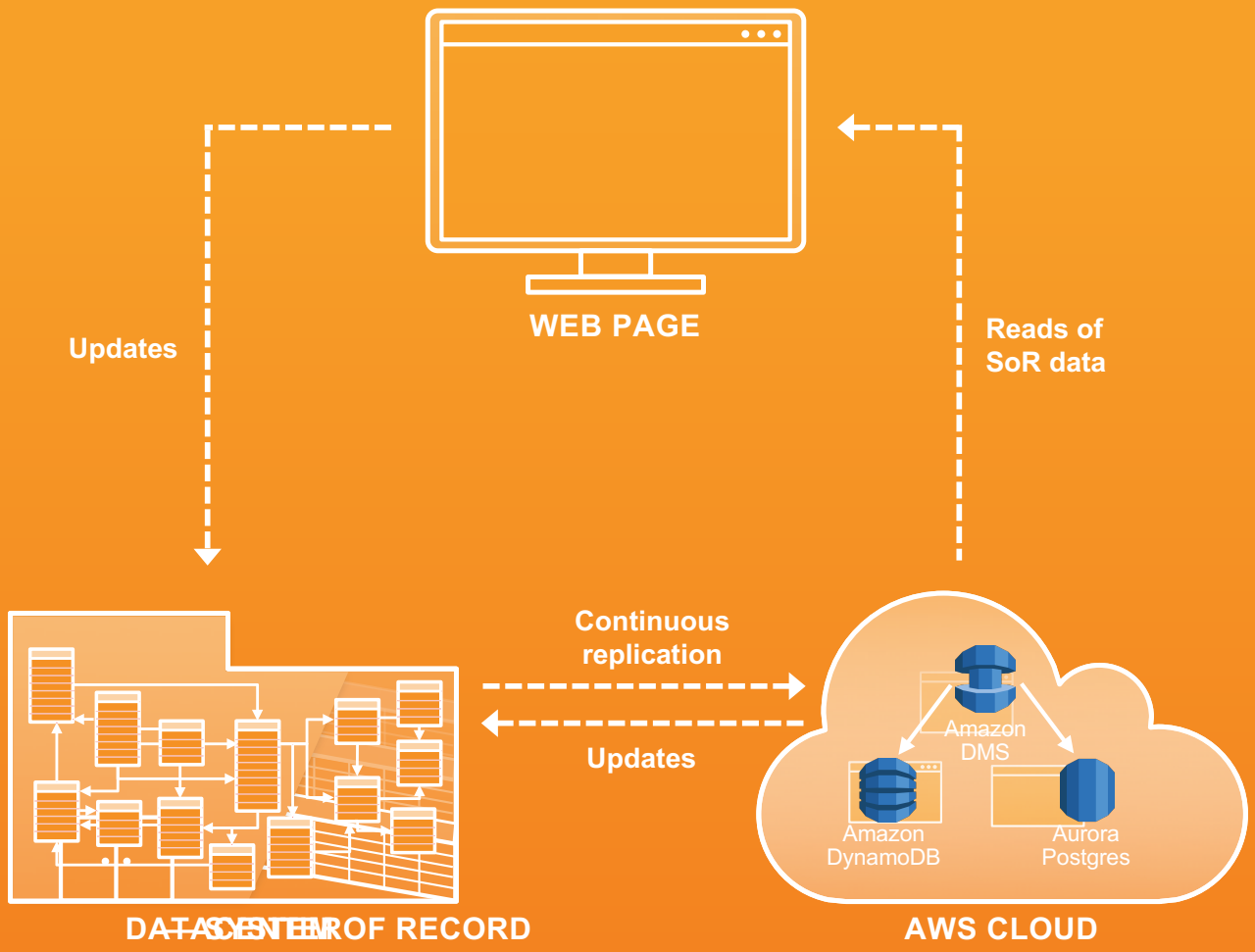
Gradual Migration



How to Run Both?

AWS Database Migration Service

Move from Oracle to scalable low cost cloud database services



How to
Move Data? a?

For cloud to be used as the system of record an archive backup mechanism is needed

To replace offsite tape backup, create a separate account in a different region

Amazon S3 is extremely secure and durable.

Data can’t be deleted. Automatic time based purge after 90 days

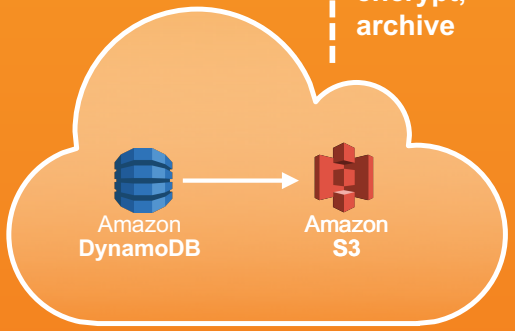
Long term very low cost archive using Amazon Glacier

How to Back up Data?

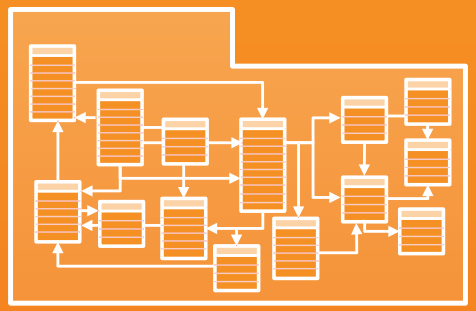
ARCHIVE
Separate AWS account,
different region

Versioned
purged after 90 days

Compress, encrypt, archive

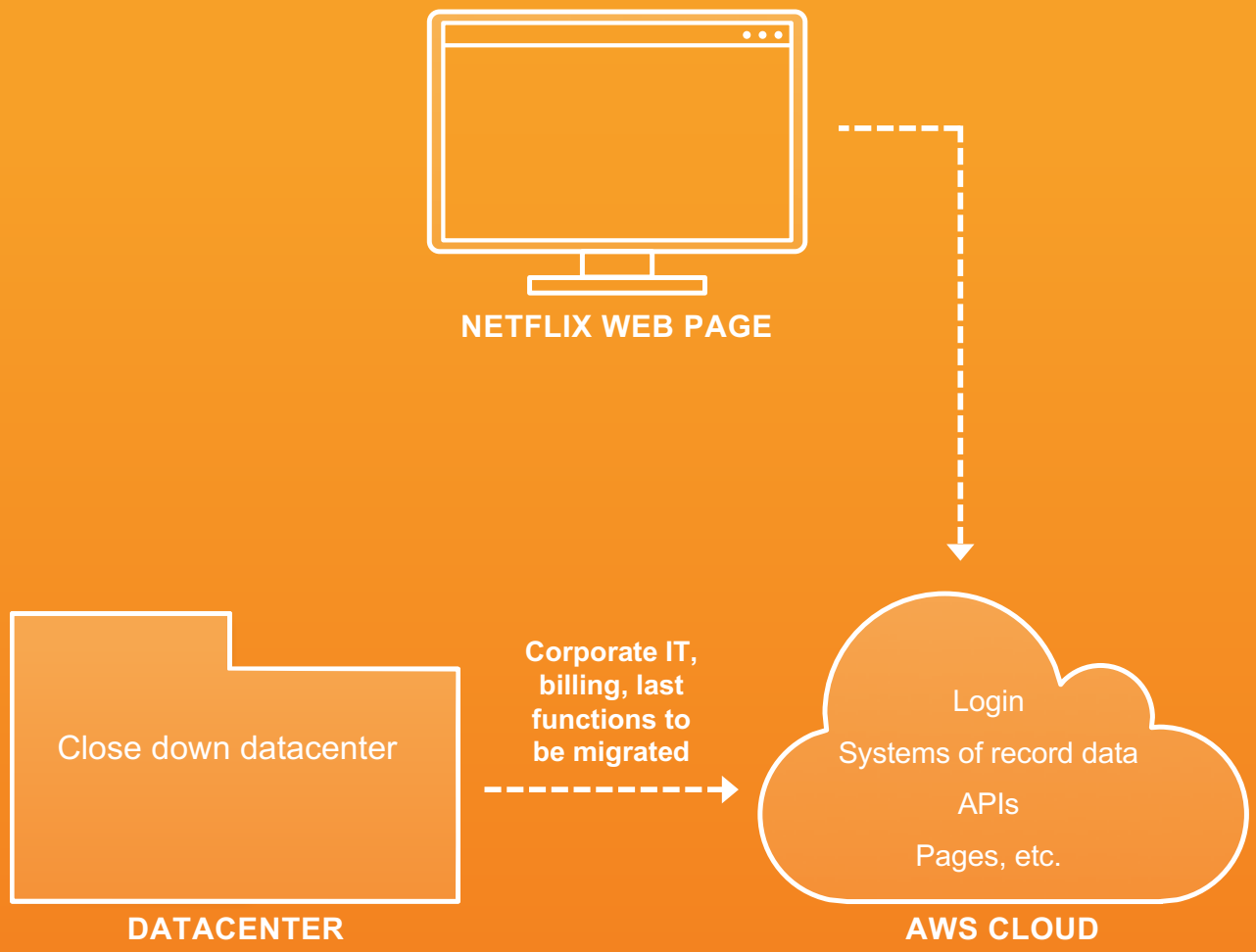


Expensive offsite tape backups



Final Stage “All-In”

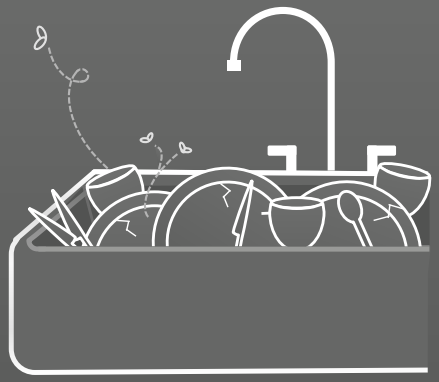
Netflix migration of billing and corporate IT



The New De-Normal

amazon web services

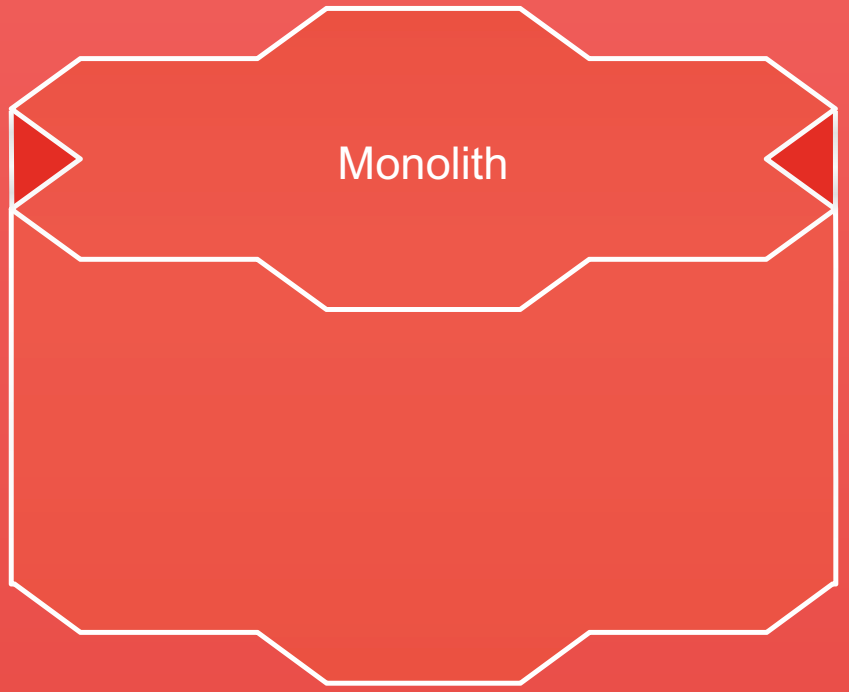
Monolithic Databases



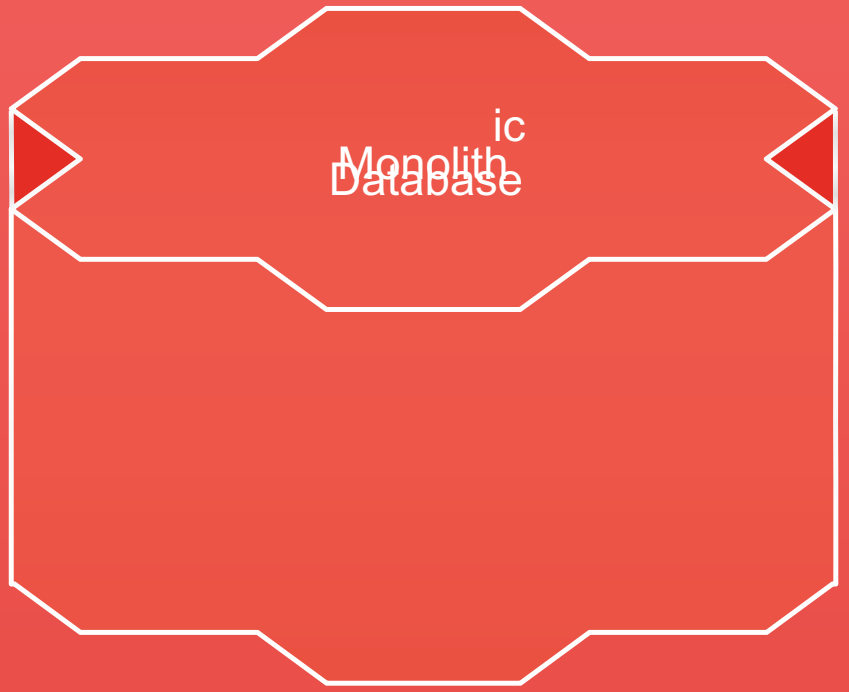
Kitchen Sink Analogy

De-normalized

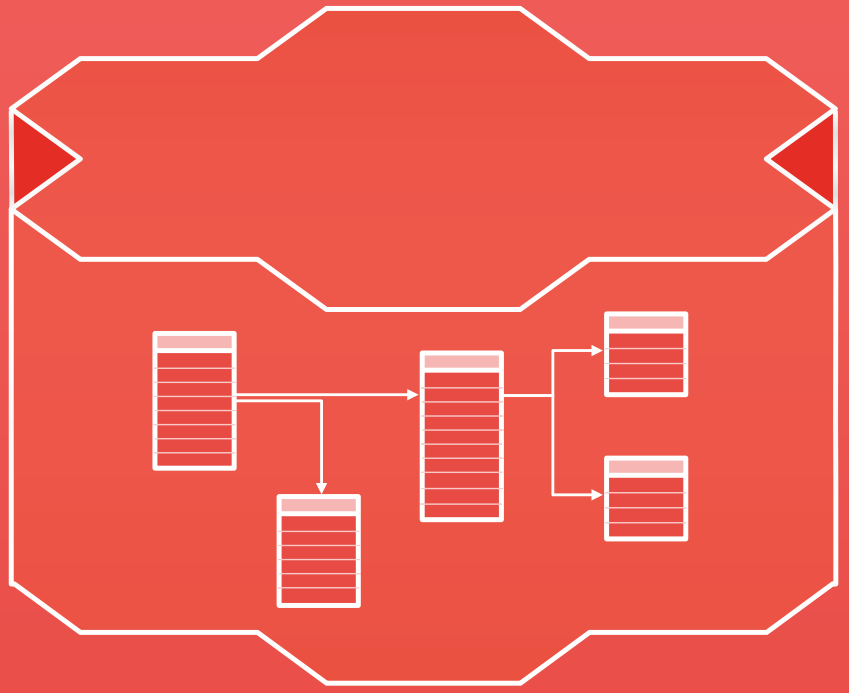
Expensive, Hard to Create and Run



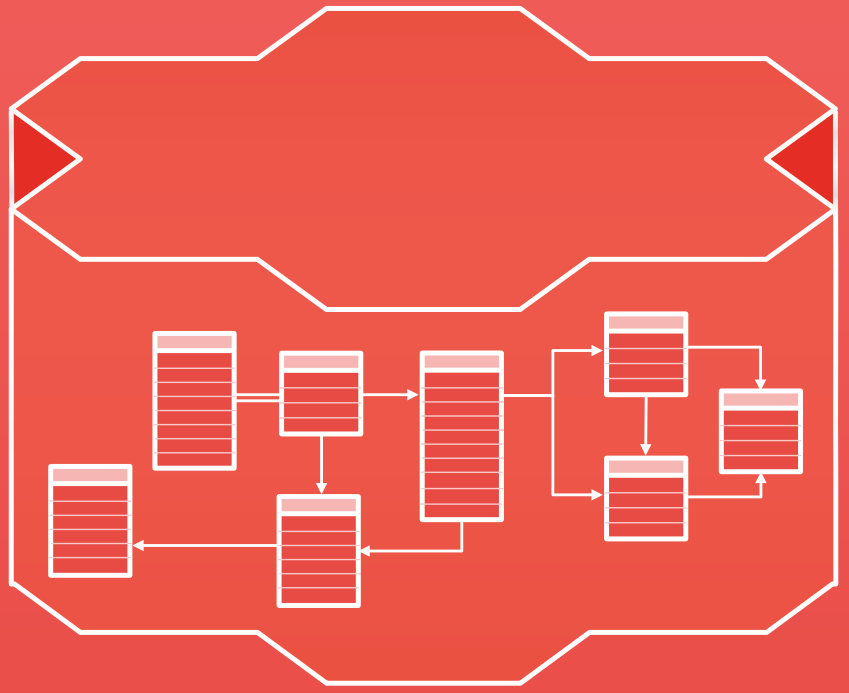
Expensive,
Hard to Create
and Run



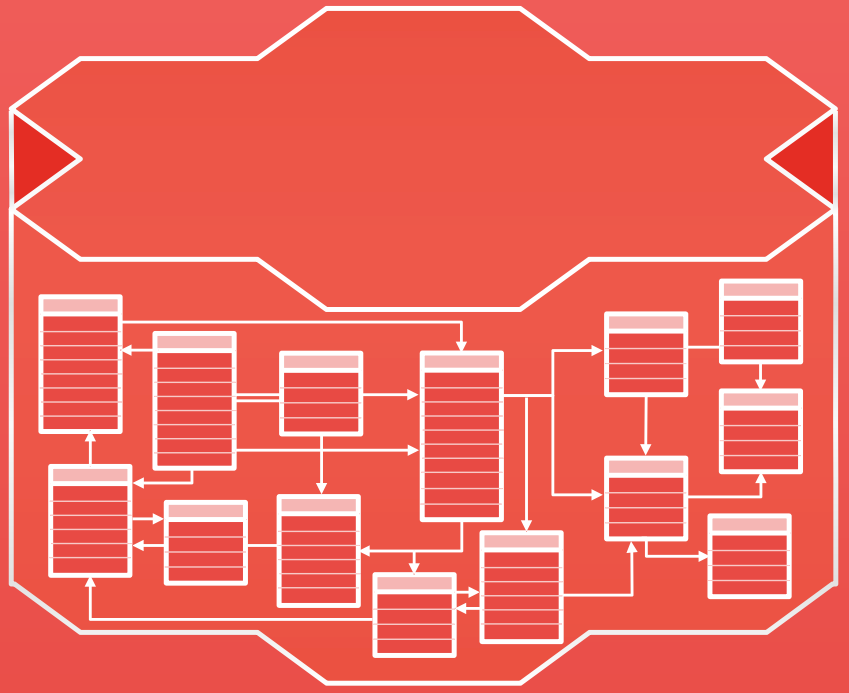
Database Schema
Entity Relationship



Database Schema
Entity Relationship



Database Schema
Entity Relationship

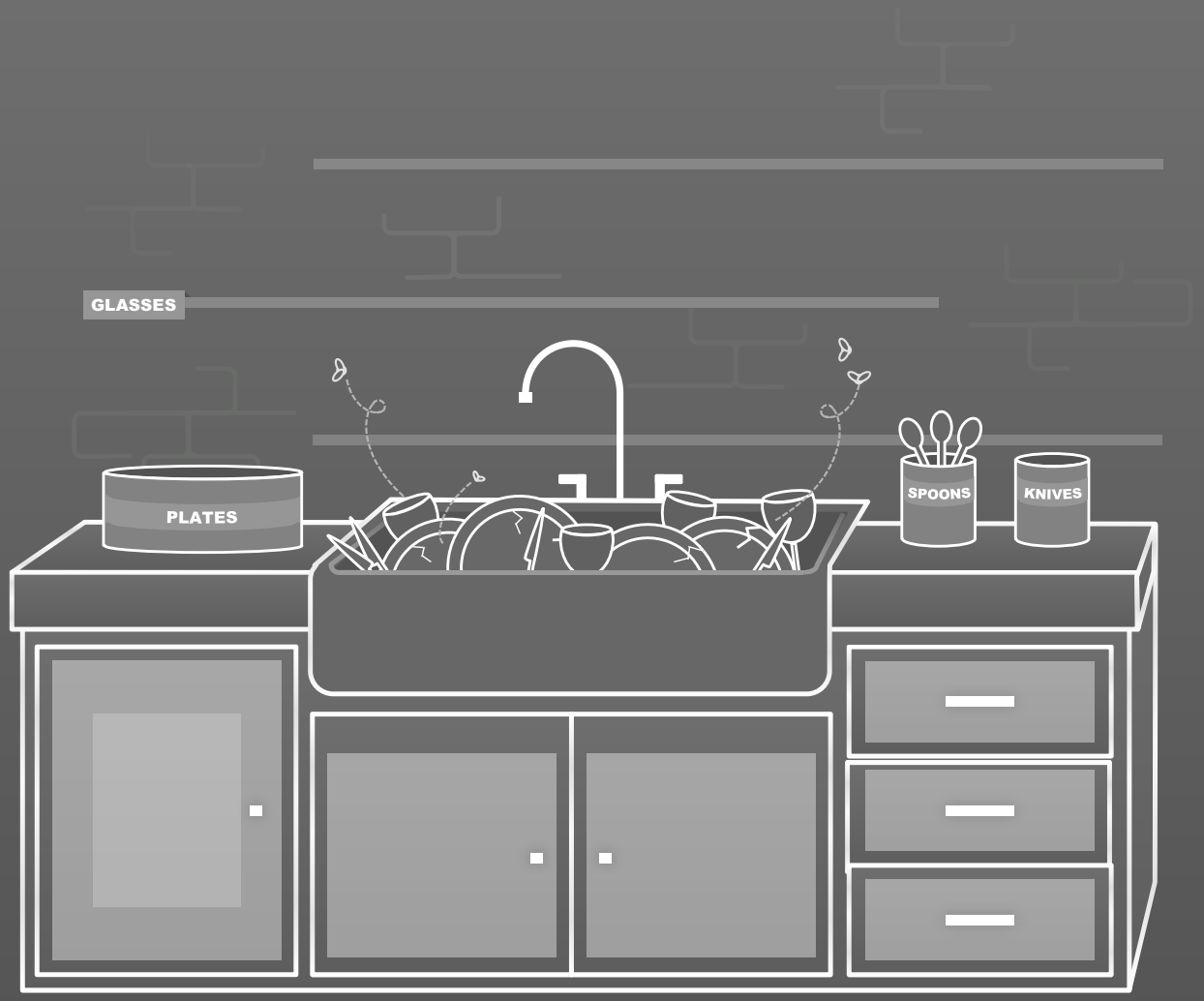


Kitchen Sink Analogy

Kitchen Sink Cleanup

Kitchen Sink Cleanup

Kitchen Sink Cleanup



Kitchen Sink Cleanup

Kitchen Sink Cleanup

Kitchen Sink Cleanup

Kitchen Sink Cleanup

Consistency Problem

How Many Complete Sets Are There?

Consistency Problem

How Many Complete Sets Are There?

Consistency Problem

How Many Complete Sets Are There?

Adding a New Use Case

Adding a New Use Case

Cloud Makes it Easy to Add New Databases

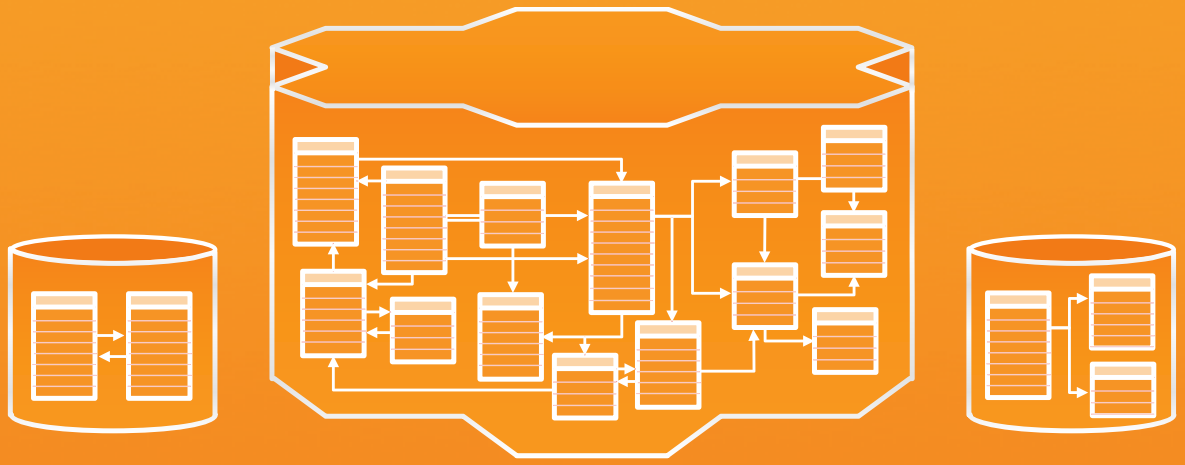
Amazon DynamoDB

Amazon DMS

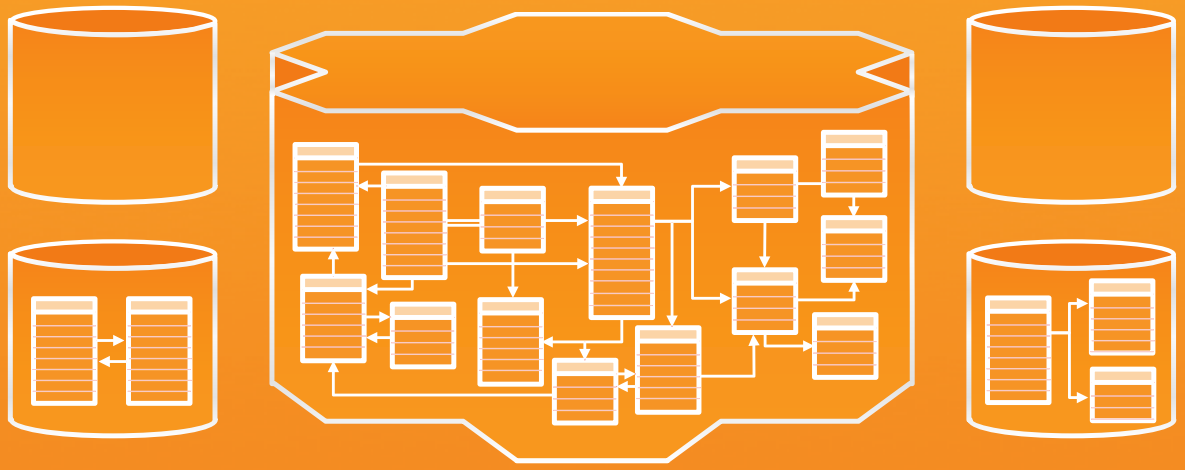
Amazon RDS

Amazon Redshift

Amazon Aurora for MySQL and Postgres



Untangle and Migrate Existing “Kitchen Sink” Schemas

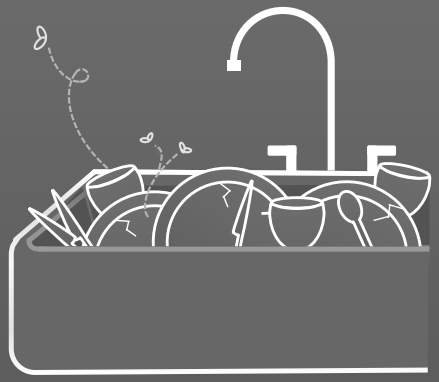


Untangle and Migrate Existing "Kitchen Sink" Schemas

The New De-Normal

amazon web services

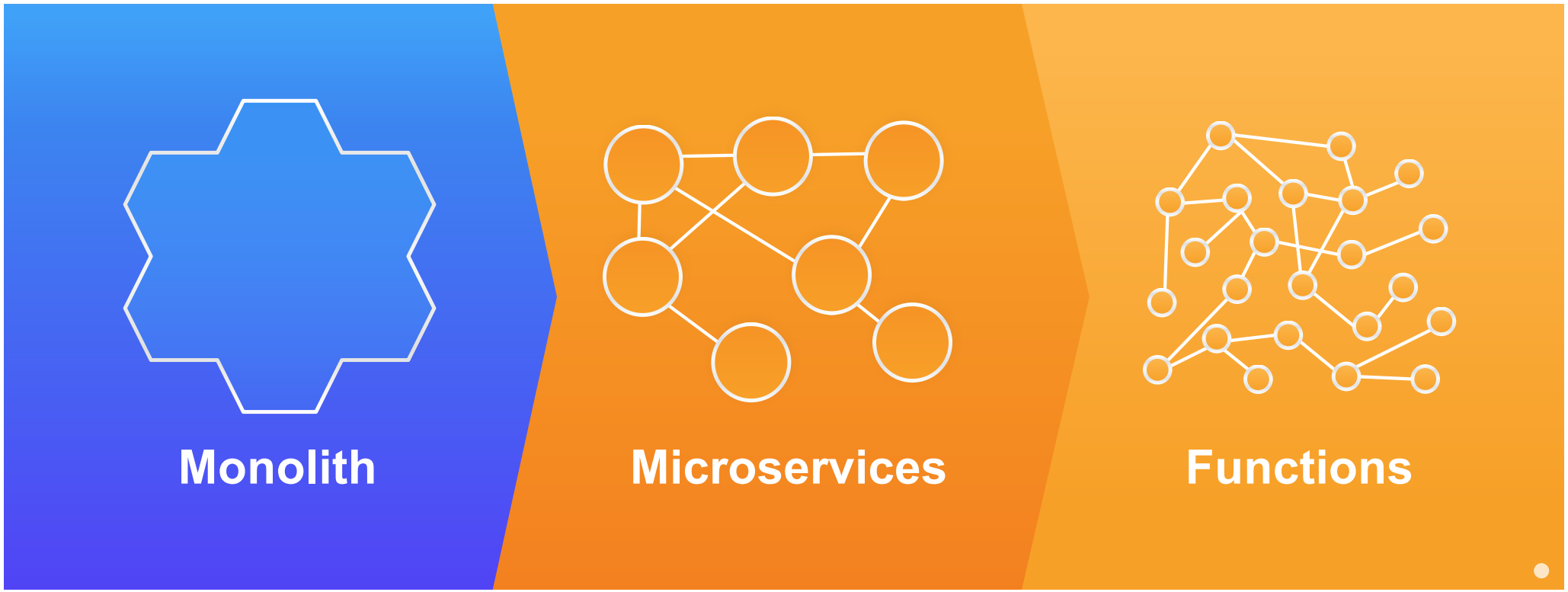
Monolithic Databases



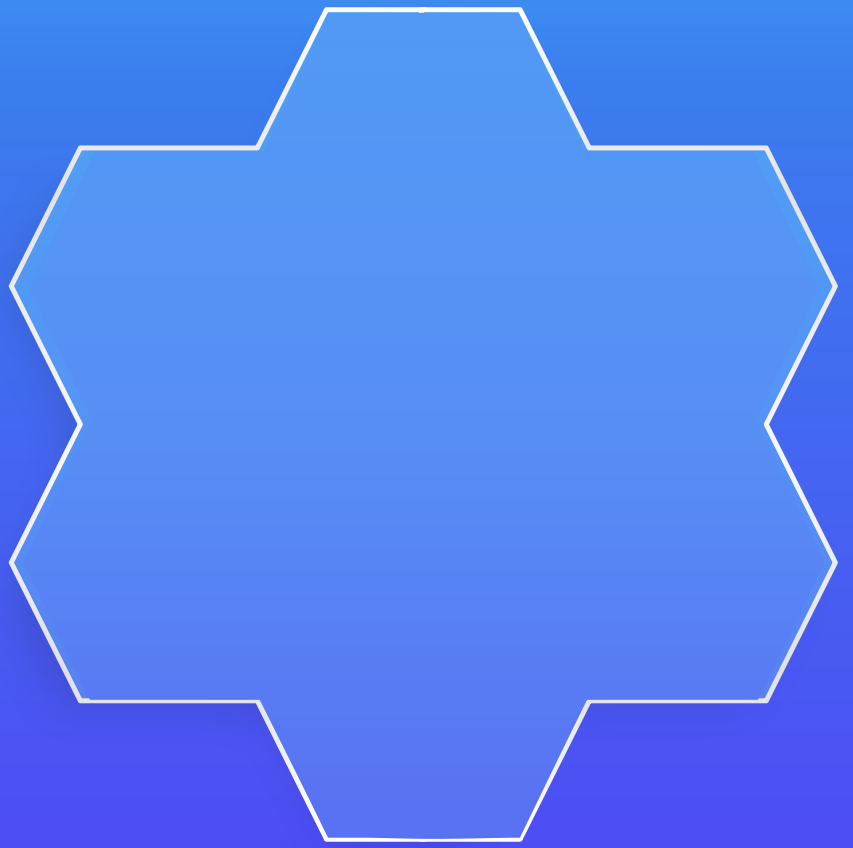
Kitchen Sink Analogy

De-normalized

Evolution of Business Logic

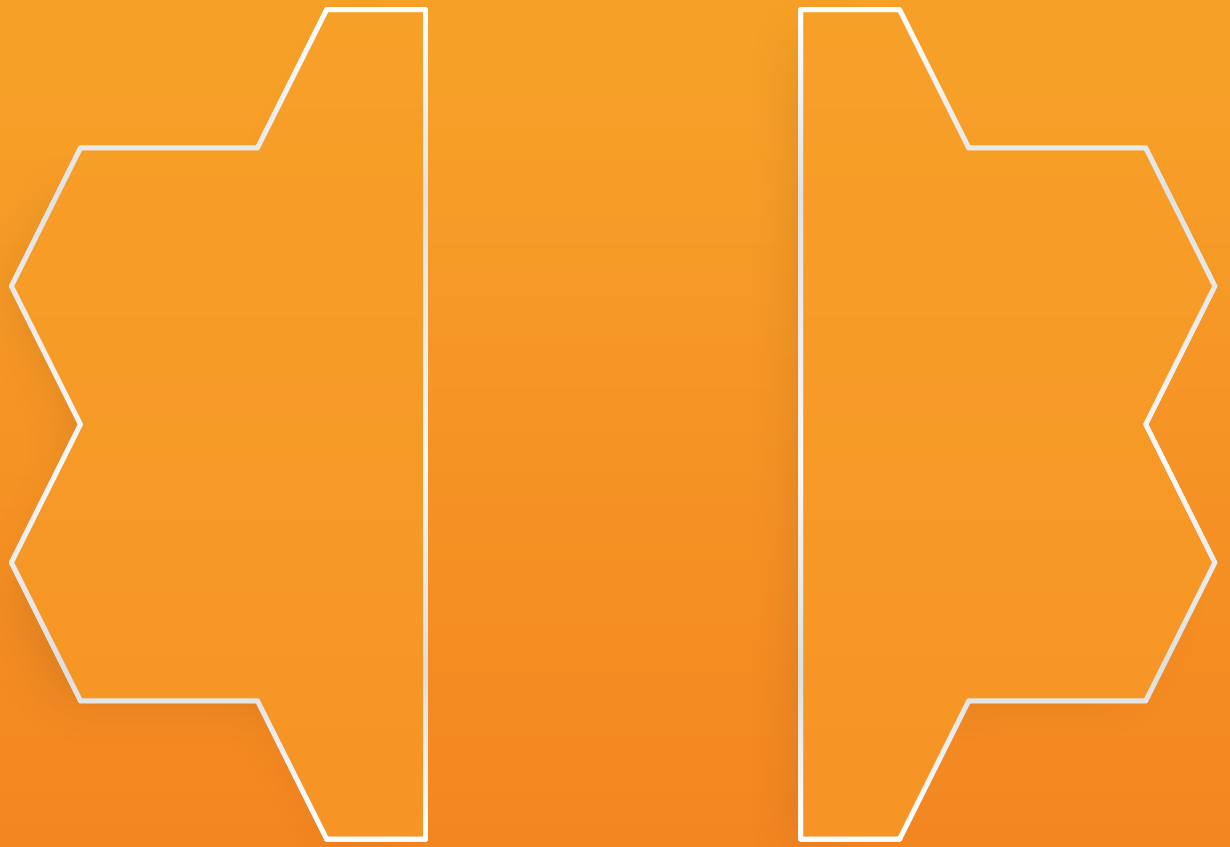


Splitting Monoliths Ten Years Ago

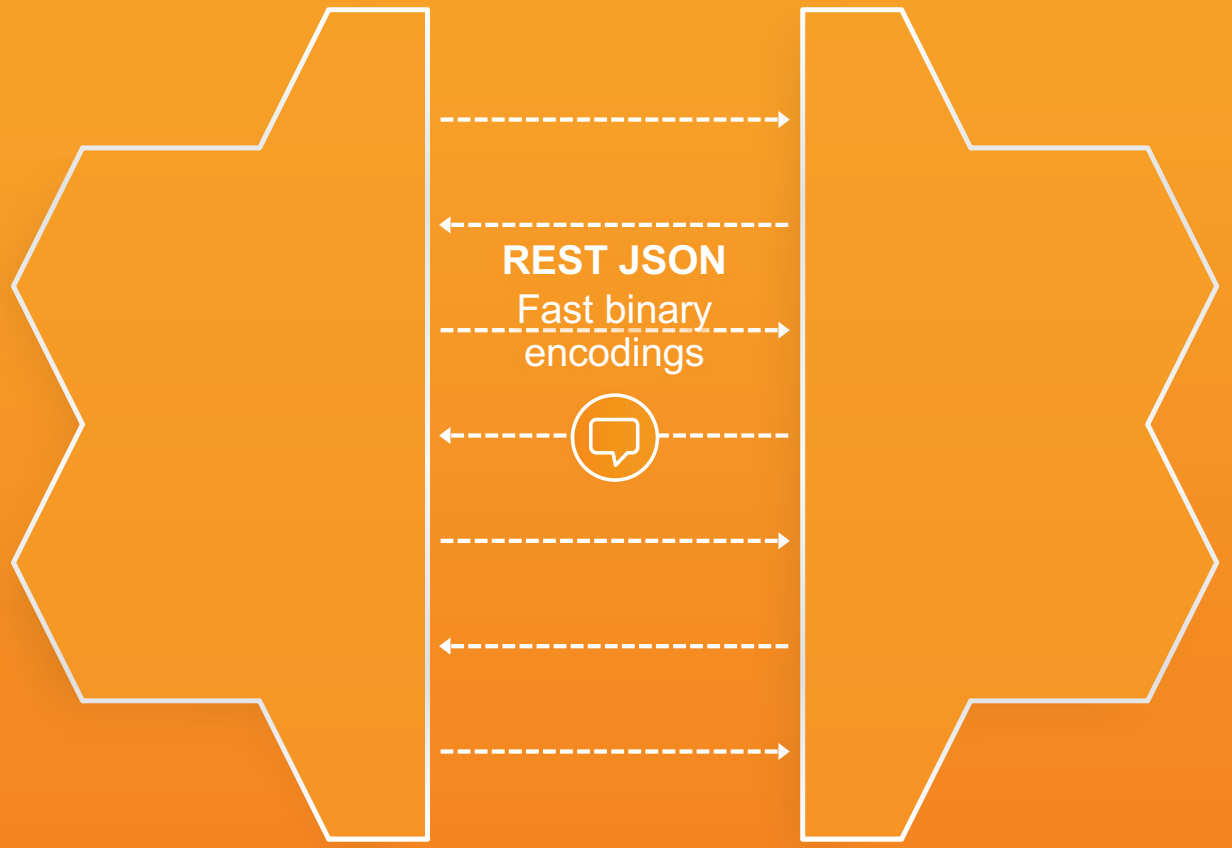


Splitting Monoliths Ten Years Ago

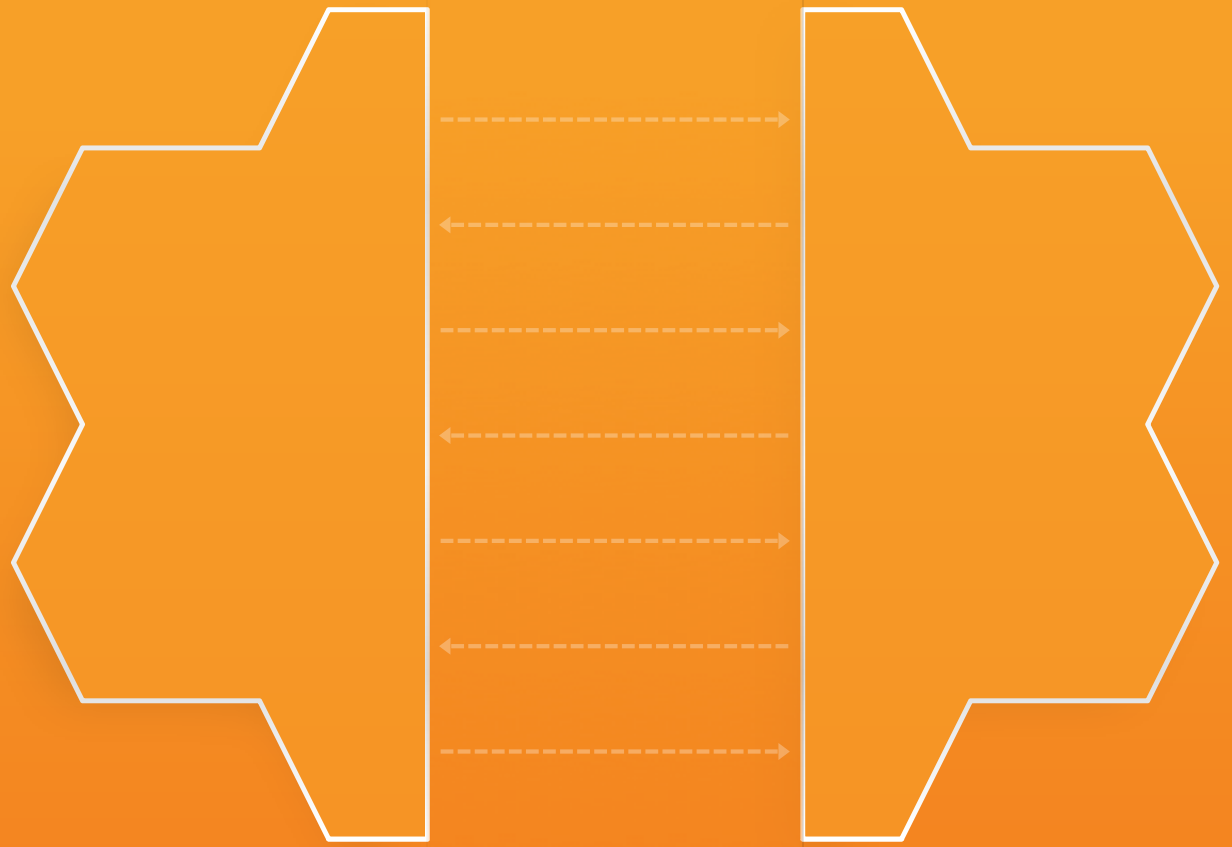
Splitting Monoliths FiveYears Ago

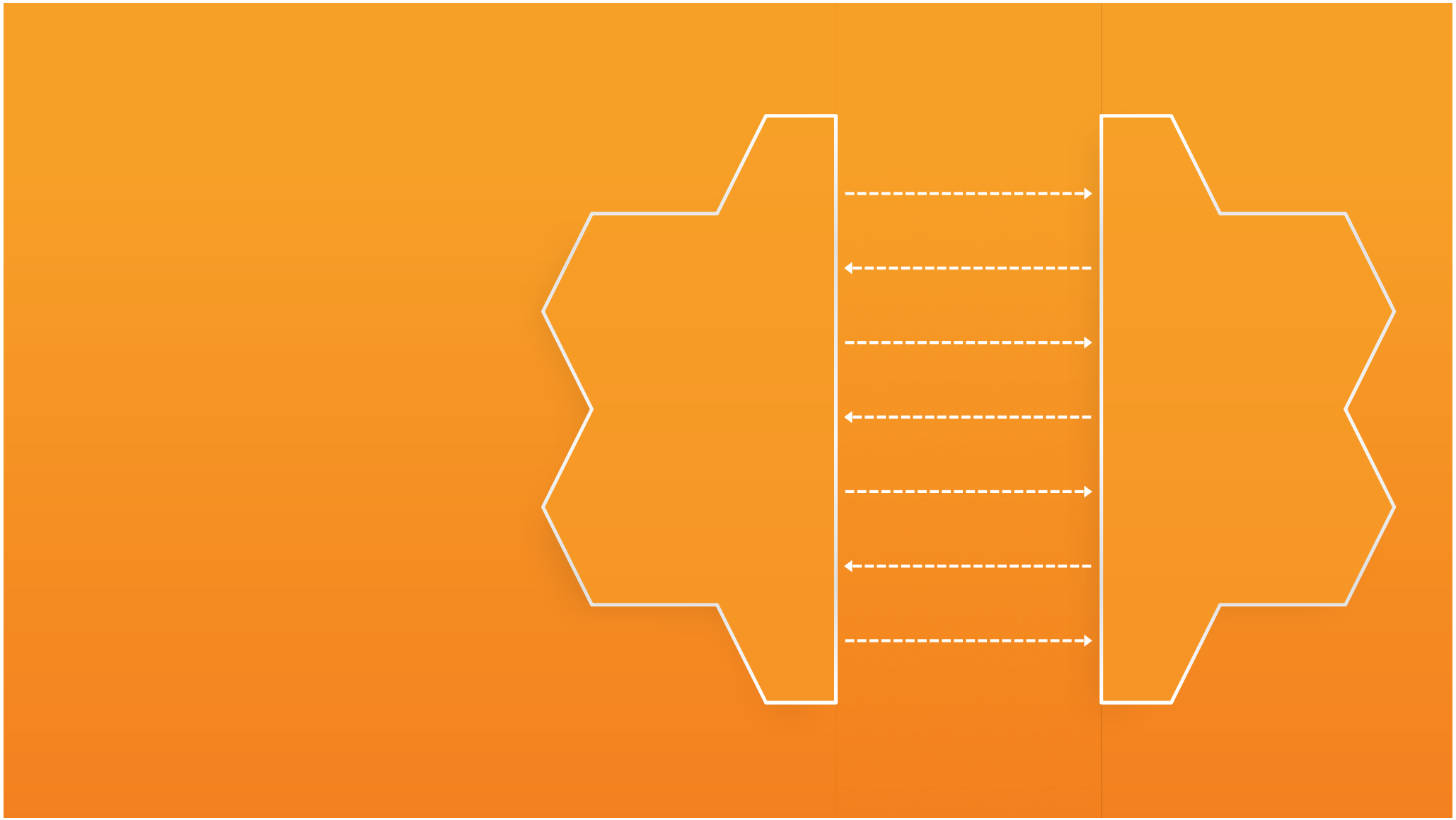


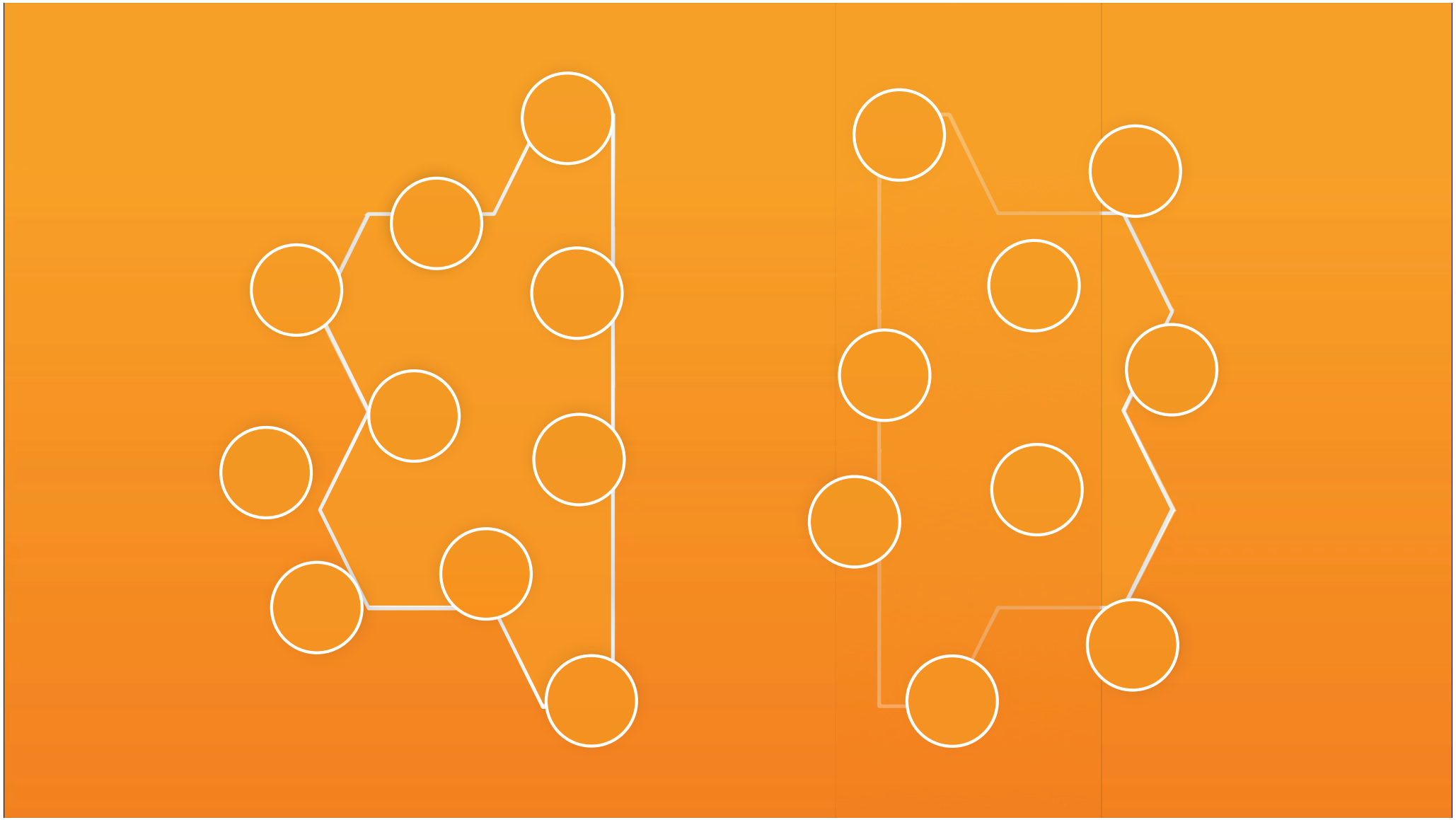
Splitting Monoliths Five Years Ago

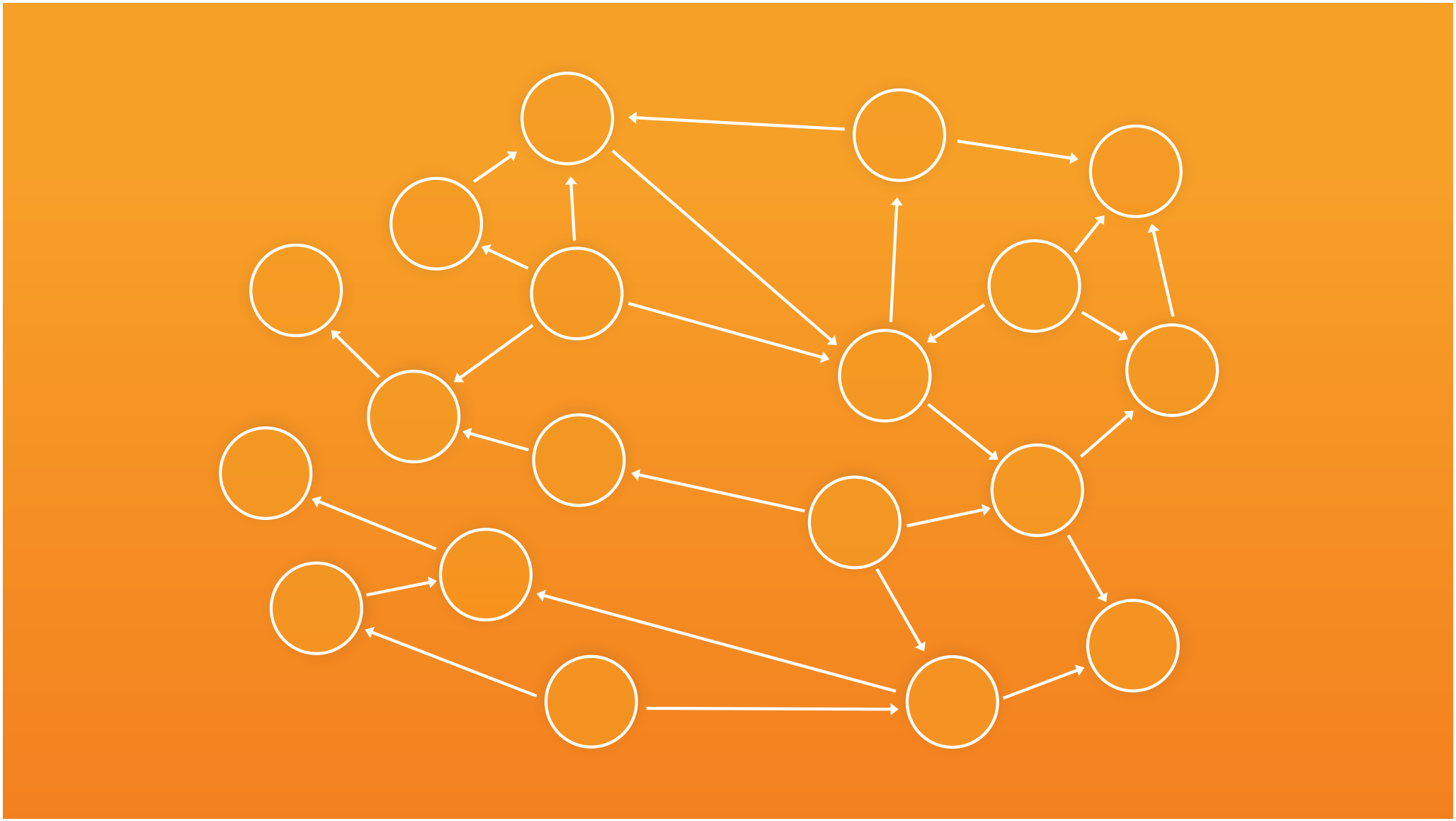


Splitting Monoliths Five Years Ago

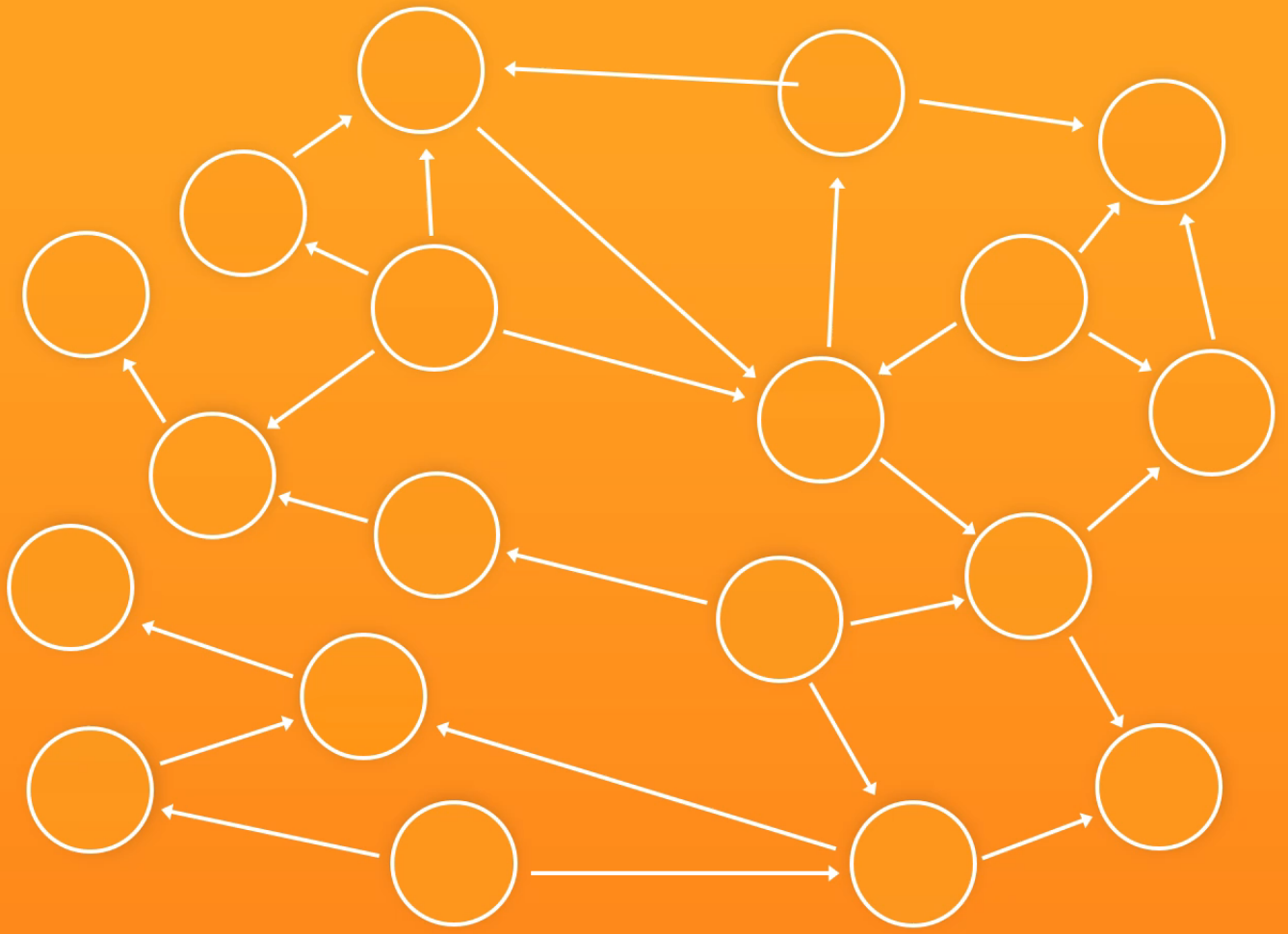








Microservices Five Years Ago



Microservices

Standard building brick services provide standardized platform capabilities

Amazon API Gateway

Amazon SQS

Amazon DynamoDB

Amazon S3

Amazon SNS

Amazon Kinesis

Microservices to Functions

Standard building brick services provide standardized platform capabilities

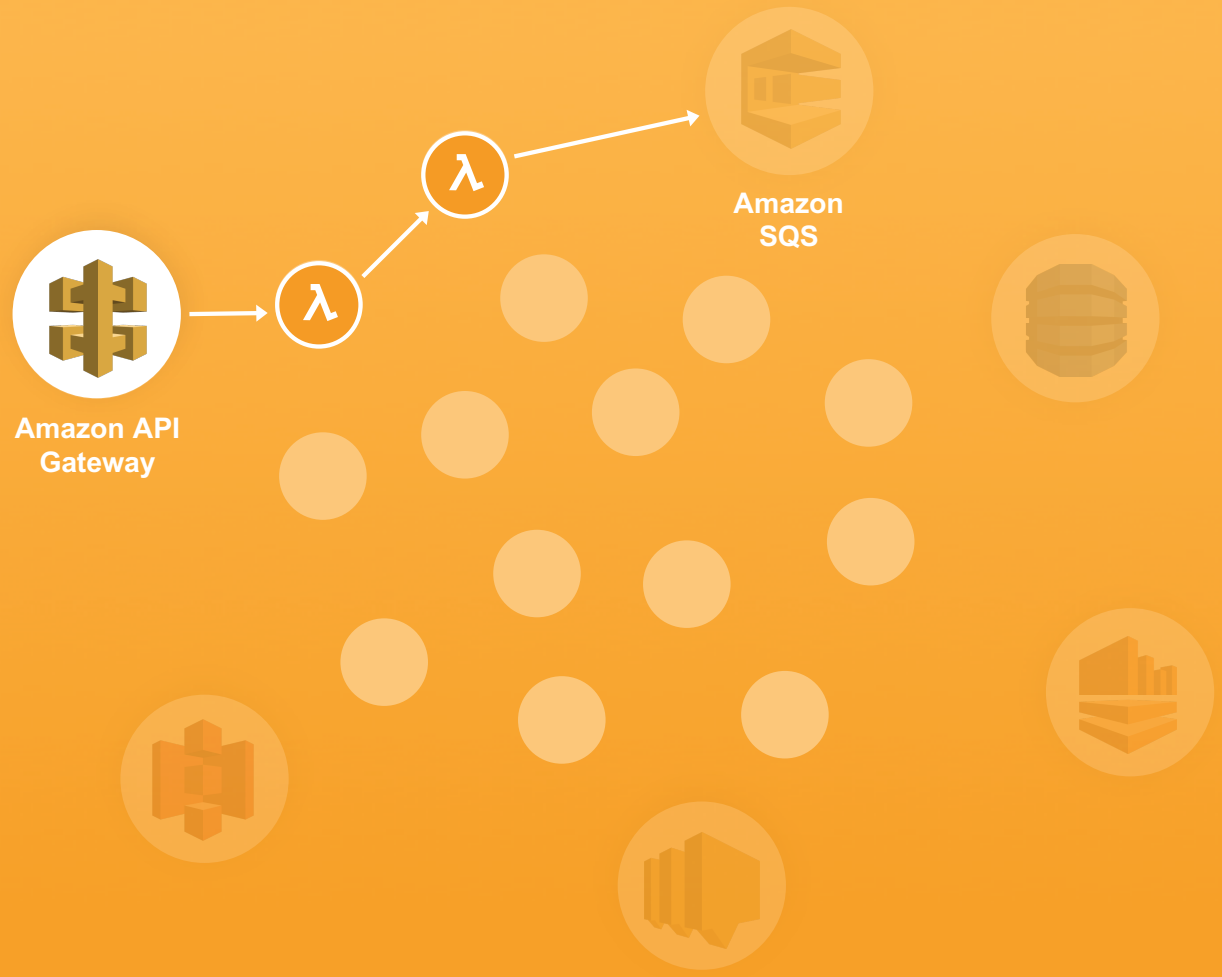
Microservices to Functions

Microservices to Functions

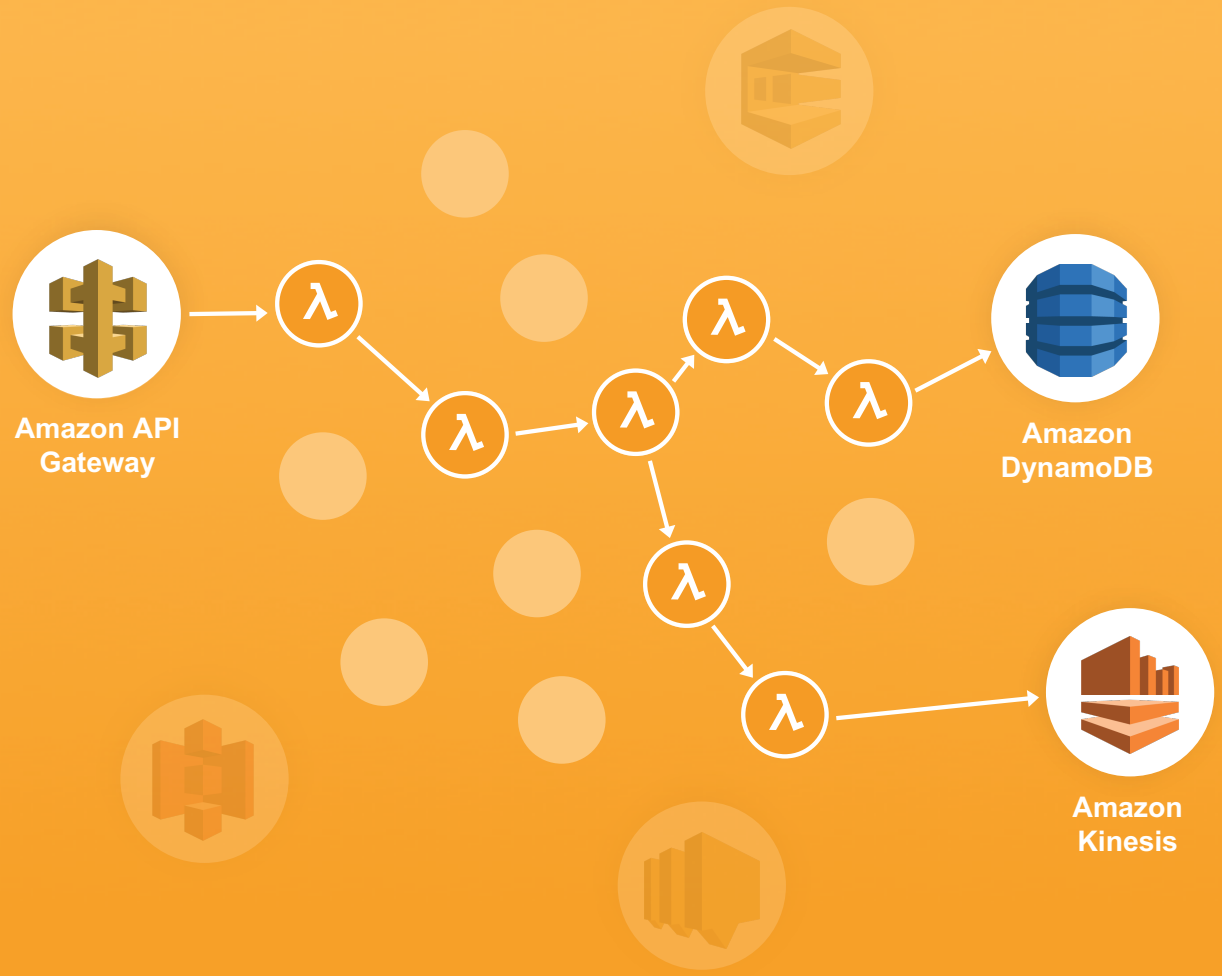
Microservices to Functions

Microservices to Ephemeral Functions

Microservices to Ephemeral Functions



Microservices to Ephemeral Functions

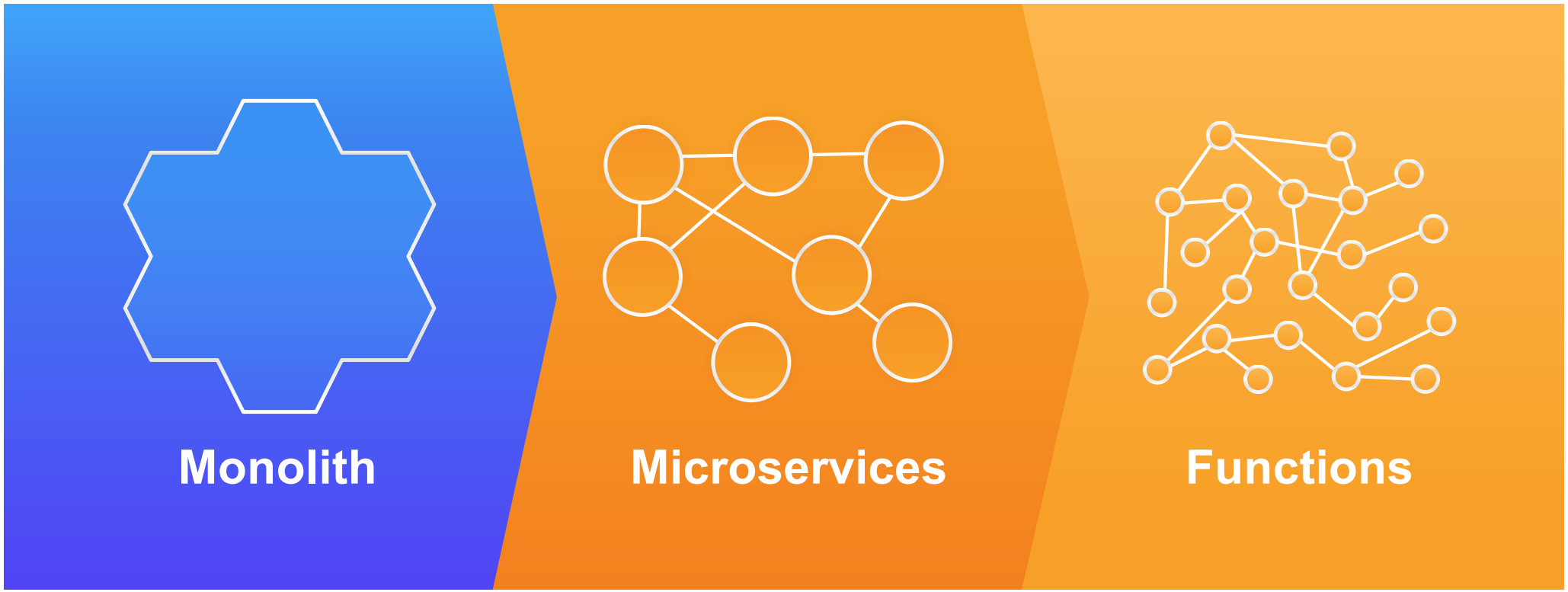


Microservices to Ephemeral Functions

Microservices to Ephemeral Functions

When the system is idle, it shuts down and costs nothing to run

Evolution of Business Logic



Open Source AI at AWS, and Apache MXNet

What I’m currently working on…

- Open Source at AWS
- Applications of Deep Learning
- Apache MXNet Overview
- Apache MXNet API
- Code and DIYrobocars
- Tools and Resources

Amazon Open Source Contributions

Linux & Drivers
Xen
Apache Tomcat
PostgreSQL
Docker
Boto

Apache Hadoop
Apache Hive
Apache Bigtop
Apache Oozie
Apache Drill
Apache Zeppelin
Apache Pig
Cloudera HUE

Apache Lucene
Apache Solr
Kuromoji
ElasticSearch
CBMC
Apache MXNet
Moses
Apache Joshua

Repositories Owned by AWS & Amazon

https://github.com/aws https://github.com/awslabs https://github.com/amznlabs
https://blox.github.io/

Blox – Container orchestration for ECS

s2n – Secure replacement for openssl, used by S3

Chalice – Python serverless microframework for AWS

Sockeye – Neural Machine Translation using MXNet 

AWS-CLI – Command line interface to AWS

AWS-shell – Autocomplete based user interface for AWS-CLI

AWS SDKs for Java, Python, PHP, Go, Ruby etc.

AWS Mobile SDKs for iOS, Android etc.

Cloud9 Ace – Cloud based interactive development editor

Cfncluster – Build and manage HPC clusters

ION – Amazon data serialization libraries

Open Source Made Easy by AWS Services

Amazon EMR – Elastic Hadoop
Amazon ElastiCache – Memcached and Redis
Amazon RDS – MySQL
Amazon RDS – PostgreSQL
Amazon Aurora – scalable back-end for MySQL & PostgreSQL
AWS OpsWorks – Chef Automate Server
Amazon ECS – Docker Orchestration
Amazon CloudSearch
Amazon Elasticsearch

Database Migration Service – Convert schemas and move data from proprietary databases to MySQL and Postgres. 22,000+ done.

Applications of Deep Learning

Why is Deep Learning Taking Off Now?

Everything is digital: large data sets are available

- Imagenet: 14M+ labeled images - http://www.image-net.org/
- YouTube-8M: 7M+ labeled videos - https://research.google.com/youtube8m/
- AWS public data sets - https://aws.amazon.com/public-datasets/

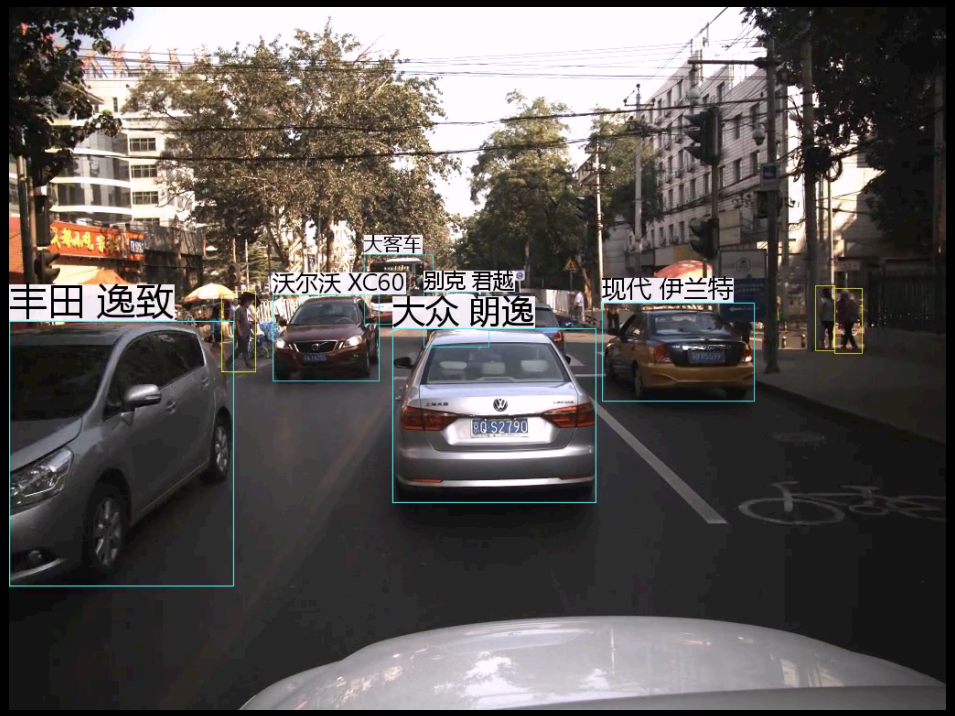
The parallel computing power of GPUs make training possible

- Simard et al (2005), Ciresan et al (2011)
- State of the art networks have hundreds of layers
- Baidu's Chinese speech recognition: 4TB of training data, +/- 10 Exaflops

Cloud scalability and elasticity make training affordable

- Grab a lot of resources for fast training, then release them
- Using a DL model is lightweight: you can do it on a Raspberry Pi

Autonomous Driving Systems



AI on AWS Today

- Zillow
  - Zestimate (using Apache Spark)
- Howard Hughes Corp
  - Lead scoring for luxury real estate purchase predictions
- FINRA
  - Anomaly detection, sequence matching, regression analysis, network/tribe analysis
- Netflix
  - Recommendation engine
- Pinterest
  - Image recognition search
- Fraud.net
  - Detect online payment fraud
- DataXu
  - Leverage automated & unattended ML at large scale (Amazon EMR + Spark)
- Mapillary
  - Computer vision for crowd sourced maps
- Hudl
  - Predictive analytics on sports plays
- Upserve
  - Restaurant table mgmt & POS for forecasting customer traffic
- TuSimple
  - Computer Vision for Autonomous Driving
- Clarifai
  - Computer Vision APIs

The Challenge For Artificial Intelligence: **SCALE**

Data

PBs of existing data

Aggressive migration

New data created on AWS

Training

Tons of GPUs

Elastic capacity

Pre-built images

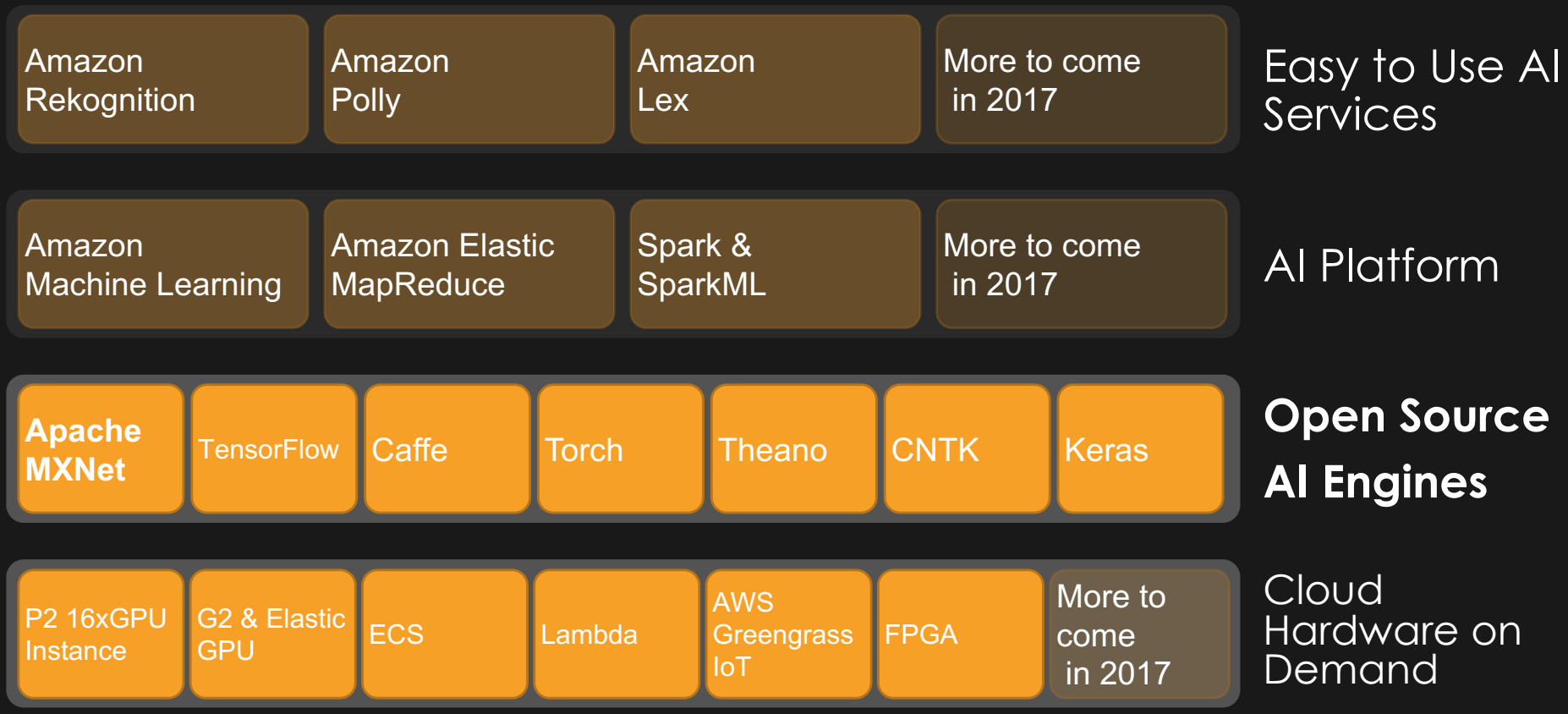
Prediction

Lots of GPUs and CPUs

Serverless

At the Edge, On IoT Devices

Amazon AI: Democratized Artificial Intelligence



AWS Deep Learning AMI

Up to~40k CUDA cores on P2

Apache MXNet

TensorFlow

Theano

Caffe & Caffe 2

Torch

Keras

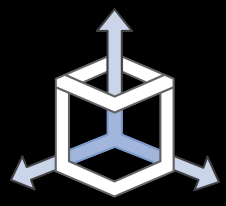
Pre-configured CUDA drivers, MKL

Anaconda, Python3

Ubuntu or Amazon Linux

+ CloudFormation template

+ Container Image



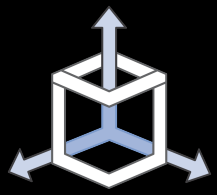
One-Click GPU & CPU
Open Source
Deep Learning
Installed, Tested, Tuned
Bootable Machine Image

Apache MXNet Overview

Apache MXNet

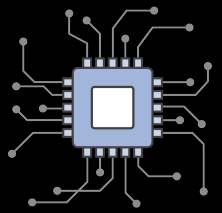
Programmable

Simple syntax,
multiple languages



Portable

Highly efficient
models for mobile
and IoT



High Performance

Near linear scaling
across hundreds of GPUs

Open Governance

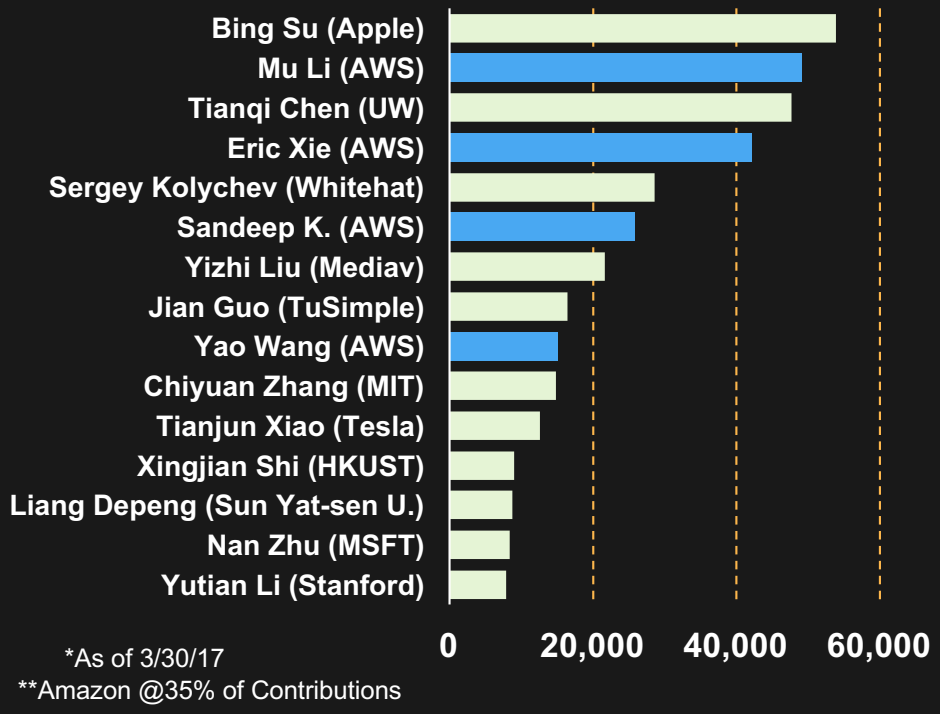
Accepted into the
Apache Incubator

Tuned On AWS

Optimized performance and
scalability on AWS GPUs

Apache MXNet | Collaborations and Community

Diverse Community



*As of 3/30/17
**Amazon @35% of Contributions

| Amazon Contributions 35%

https://wiki.apache.org/incubator/May2017

In the last month, excluding merges, 51 authors have pushed 165 commits to master and 180 commits to all branches.

Deep Learning Framework Comparison

	Apache MXNet	**TensorFlow**	**Cognitive Toolkit**
Industry Owner	N/A – Apache Community	Google	Microsoft
Programmability	Imperative and Declarative	Declarative only	Declarative only
Language Support	R, Python, Scala, Julia, C++. Javascript, Go, Matlab, Perl...	Python, C++. Experimental Go and Java	Python, C++, Brainscript.
Code Length | AlexNet (Python)	44 sloc	107 sloc using TF.Slim	214 sloc
Memory Footprint (LSTM)	2.6GB	7.2GB	N/A

Apache MXNet | Amazon Strategy

Integrate with AWS Services

Bring Scalable Deep Learning to EMR, Lambda, ECS and many more..

Foundation for AI Services

Higher Velocity for AI Services, Research and Core AI Development

Leverage the Community

Community brings velocity and innovation with no industry ownership
Safest for long term investment

Deep Learning using MXNet @Amazon

- Applied Research
- Core Research
- Alexa
- Demand Forecasting
- Risk Analytics
- Search
- Recommendations
- AI Services | Rek, Lex, Polly
- Q&A Systems
- Supply Chain Optimization
- Advertising
- Machine Translation
- Video Content Analysis
- Robotics
- Lots of Computer Vision..
- Lots of NLP/U..

*Teams are either actively evaluating, in development, or transitioning to scale production

Apache MXNet API

Apache MXNet | The Basics

- ***NDArray*:** Manipulate multi-dimensional arrays (tensors) in a command line paradigm (imperative).
- ***Symbol*:** Symbolic expression for neural network flows (declarative).
- ***Module***: Intermediate-level and high-level interface for neural network training and inference.
- **Loading Data**: Feeding data into training/inference programs.
- **Mixed Programming**: Training algorithms developed using *NDArrays* in concert with *Symbols*.

https://medium.com/@julsimon/an-introduction-to-the-mxnet-api-part-1-848febdcf8ab

Imperative Programming

```
import numpy as np
a = np.ones(10)
b = np.ones(10) * 2
c = b * a
d = c + 1
```

Easy to tweak
in Python, R, Perl etc.

PROS

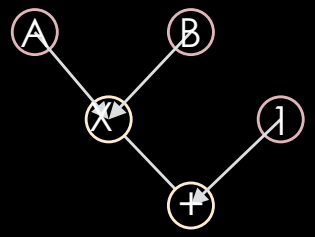
- Straightforward and flexible.
- Take advantage of language native features (loop, condition, debugger).
- E.g. Numpy, Matlab, Torch, …

CONS

- Hard to optimize

Declarative Programming

```
A = Variable('A')
B = Variable('B')
C = B * A
D = C + 1
f = compile(D)
d = f(A=np.ones(10),
      B=np.ones(10)*2)
```



C can share memory with D because C is deleted later

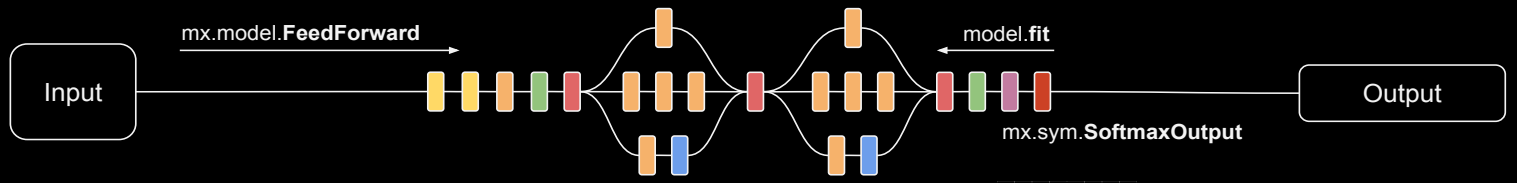
PROS

- More chances for optimization
- Separates flow structure
- E.g. TensorFlow, Theano, Caffe

CONS

- Less flexible, hard to debug

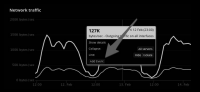
Deep Learning Models




Image


Video

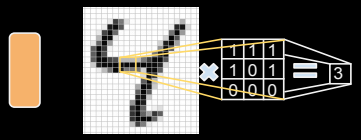

Speech


Events

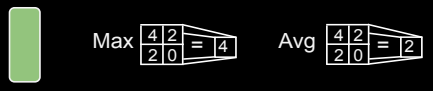
"People Riding Bikes"

Text

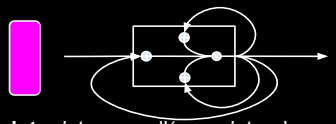
mx.sym.**FullyConnected**(data, num_hidden=128)



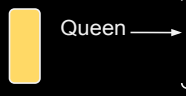
mx.sym.**Convolution**(data, kernel=(5,5), num_filter=20)



mx.sym.**Pooling**(data, pool_type="max", kernel=(2,2), stride=(2,2)



lstm.lstm_unroll(num_lstm_layer, seq_len, len, num_hidden, num_embed)



$cos(w, \textbf{queen}) = cos(w, \textbf{king}) - cos(w, \textbf{man}) + cos(w, \textbf{woman})$

mx.symbol.Embedding(data, input_dim, output_dim = k)

mx.sym.**Activation**(data, act_type="xxxx")


Image Segmentation


Face Search


Neural Art

"People Riding Bikes"

Image Caption

Bicycle, People, Road, Sport

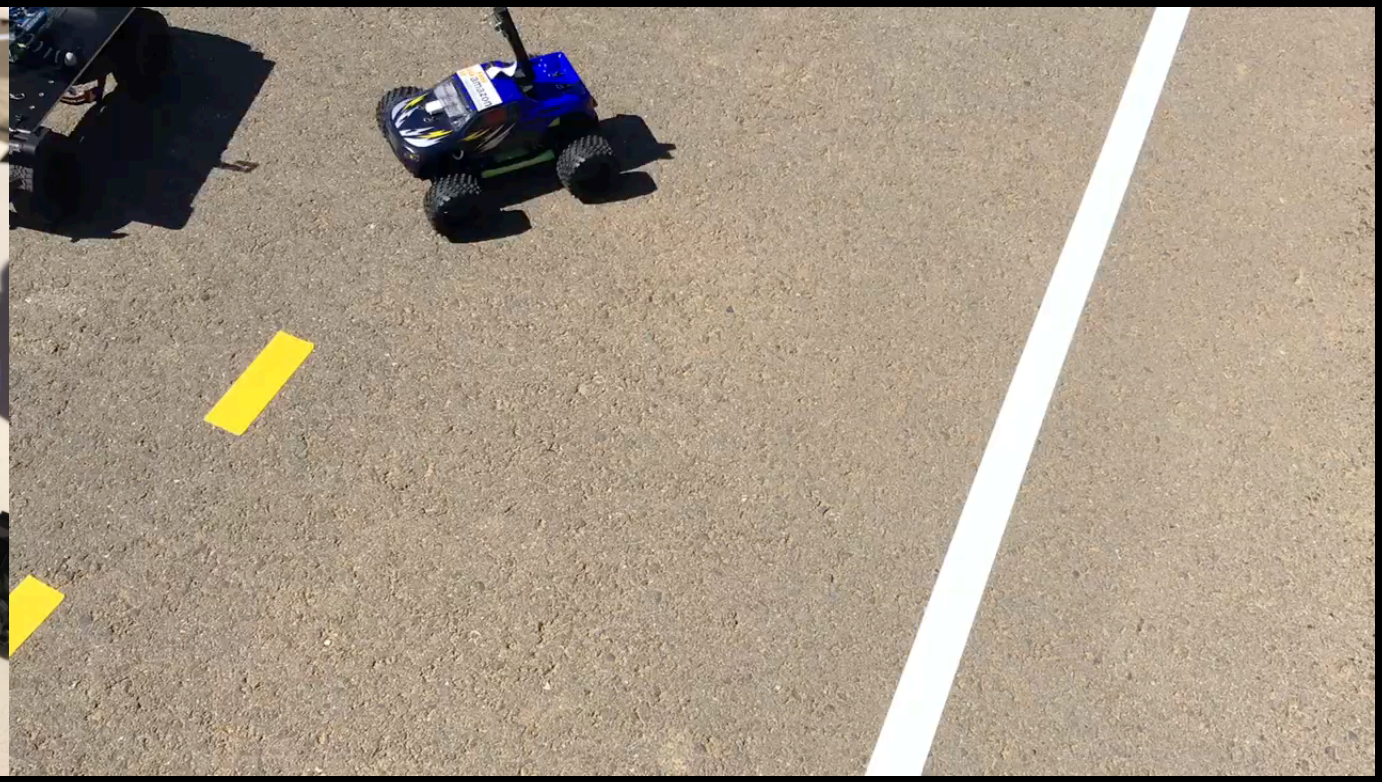
Image Labels

"Οι άνθρωποι ιπτασίας ποδήλατα"

Machine Translation

Adrian's @DIYrobocar Project – RC Truck+Raspberry Pi+EC2

Camera feeds Donkey software with Keras or MXNet model, trained on EC2 instance
Will Roscoe wrote Donkey, Sunil Mallya added MXNet, Adrian built Donkey Hoté
https://github.com/wroscoe/donkey



Additional Resources

MXNet Resources

- MXNet Blog Post | AWS Endorsement
  - http://www.allthingsdistributed.com/2016/11/mxnet-default-framework-deep-learning-aws.html
- Read up on MXNet and Learn More:
  - mxnet.io https://github.com/dmlc/mxnet/
- Re:Invent MXNet Recommender Systems Talk by Leo Dirac
  - https://www.portal.reinvent.awsevents.com/connect/sessionDetail.ww?SESSION_ID=8591

AWS Resources: follow Julien Simon @julsimon, Sunil Mallya @sunilmallya

- Deep Learning AMI
  - https://aws.amazon.com/marketplace/pp/B01M0AXXQB | Amazon Linux
  - https://aws.amazon.com/marketplace/pp/B06VSPXKDX | Ubuntu
- CloudFormation Template Instructions
  - https://github.com/dmlc/mxnet/tree/master/tools/cfn
- Deep Learning Benchmark
  - https://github.com/awslabs/deeplearning-benchmark
- MXNet on Lambda
  - https://github.com/awslabs/mxnet-lambda
- MXNet on ECS/Docker
  - https://github.com/awslabs/ecs-deep-learning-workshop

Thank You!

Adrian Cockcroft @adrianco

Animations by Silver Fox