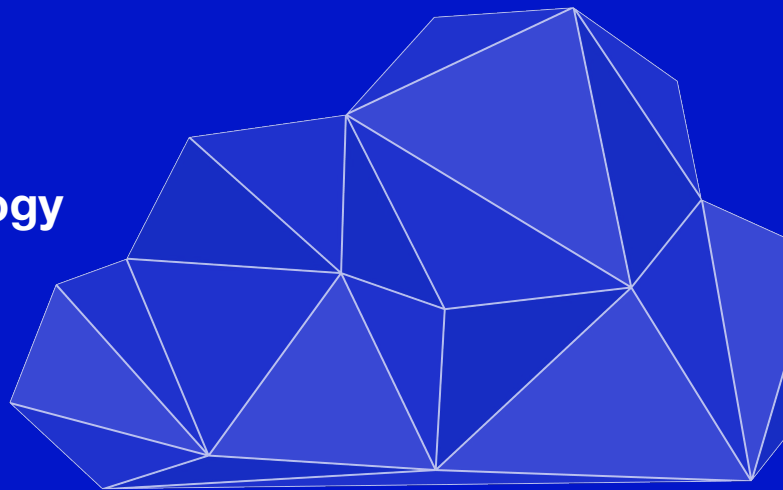


S3DMap

S3 Prefix-Level Cost Optimization Methodology
& 3D Storage Visualization Tool

Dor Azouri

VP Research @PointFive





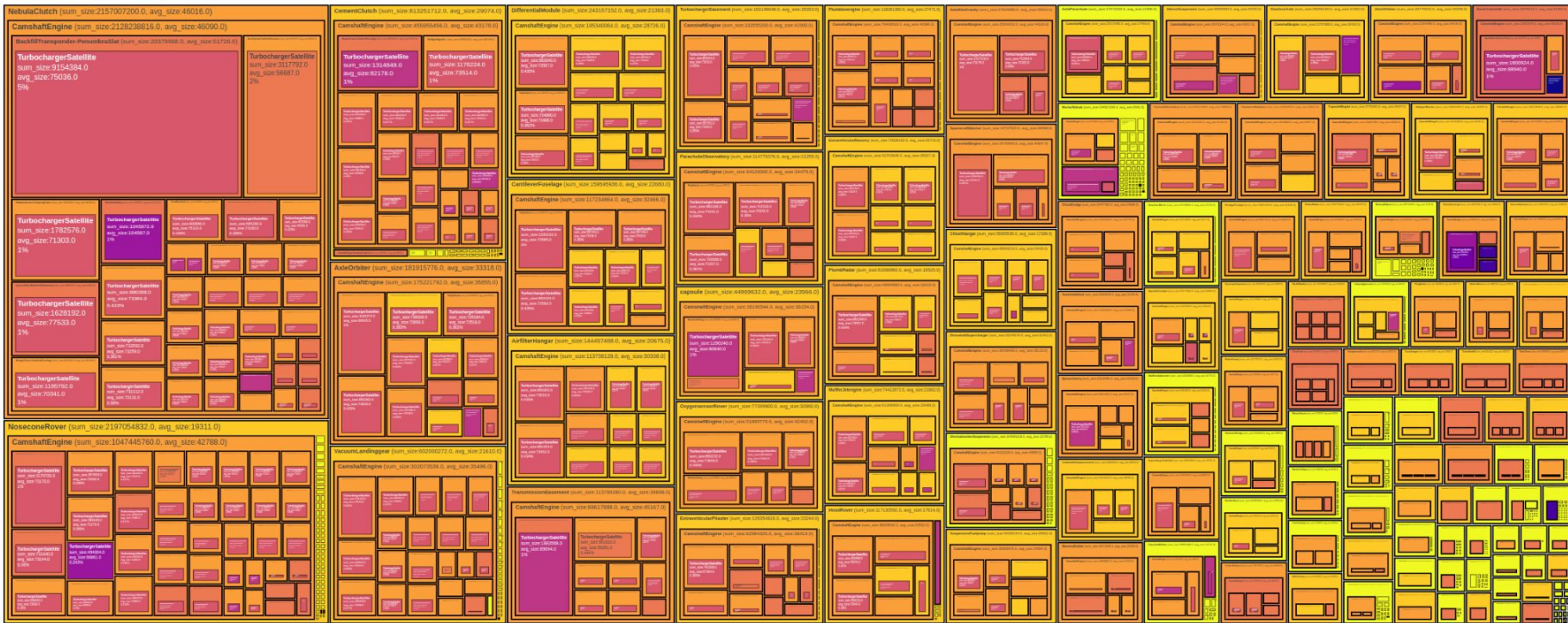
Max Depth:

Max Depth:

Sum Size

ension: Average Size

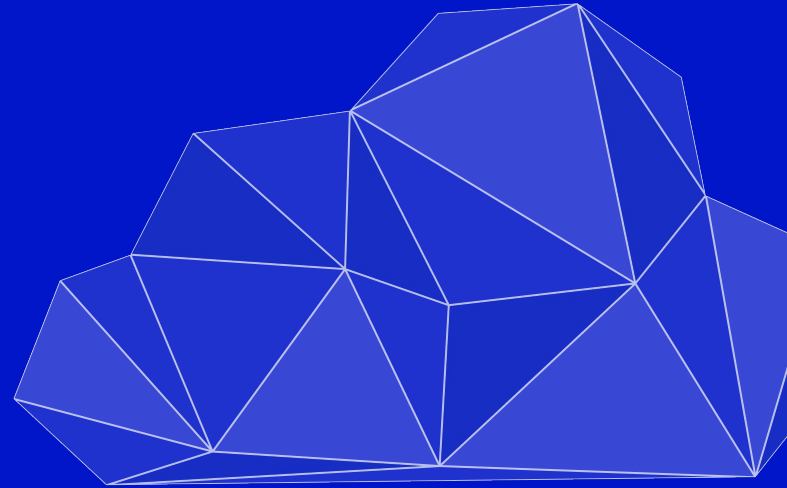
Update Treemap



<https://github.com/PointFiveLabs/s3dmap>



Agenda

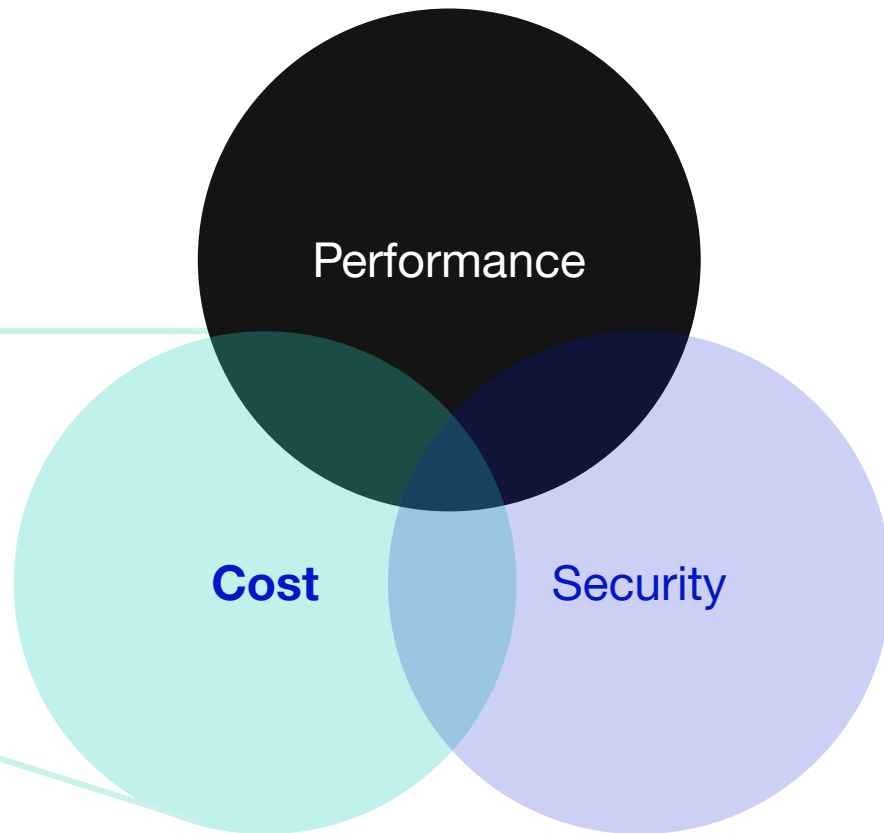


The Scope

Cost big spenders:

#1 Compute

#2 **Storage**



Talk Structure

Platform Engineering

Cost

Security

Performance

Storage

Compute

The Focus: S3 Buckets

The Challenge: Big Data & Small Letters

The Goal: Efficient Buckets Architecture

The Methodology:

Prefix-Oriented
Objects Management
(POOM)

The Tool:
S3DMap



The Ideal:

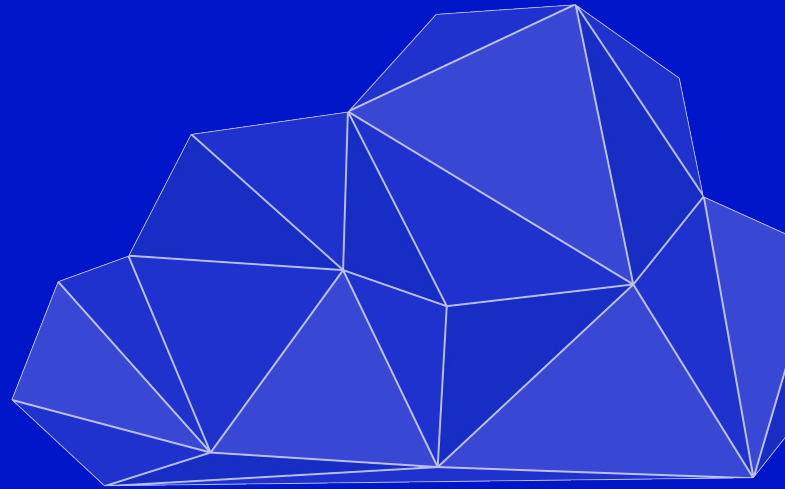
Designated
Buckets

The

Common:
Generalized
Buckets

The Focus

S3 Buckets

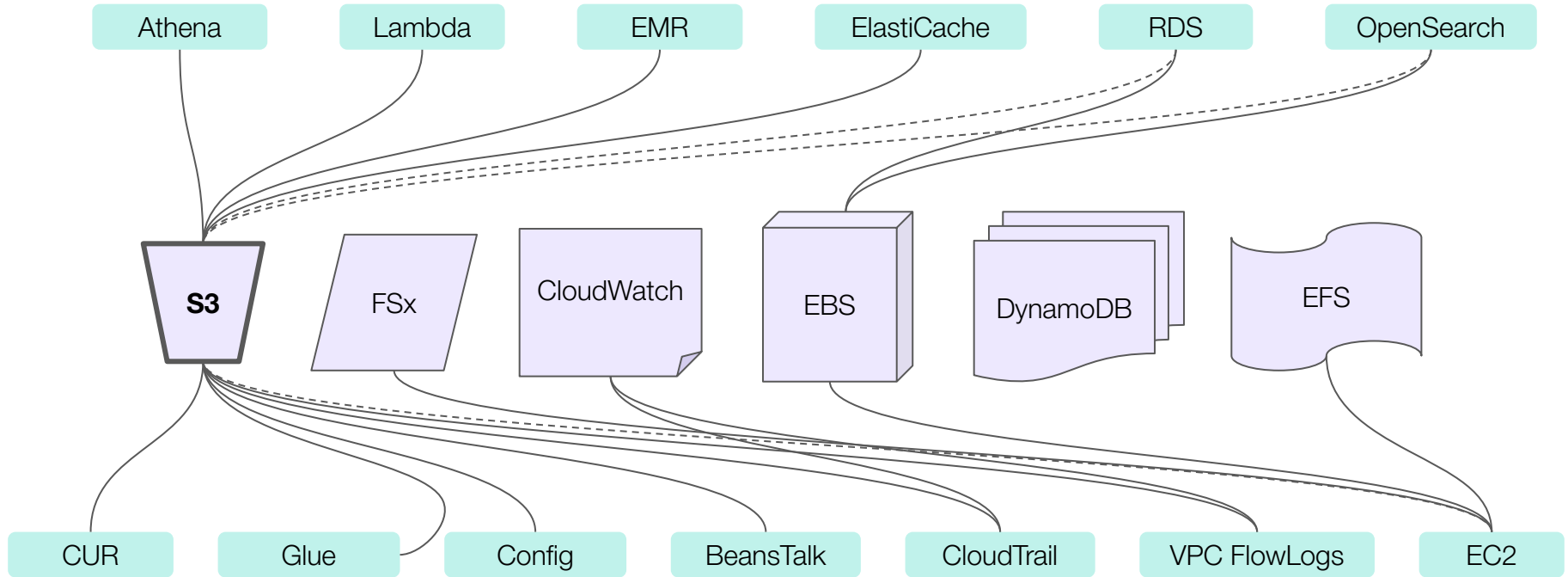


S3 General Overview

Amazon S3 is an object storage service that offers high-grade:

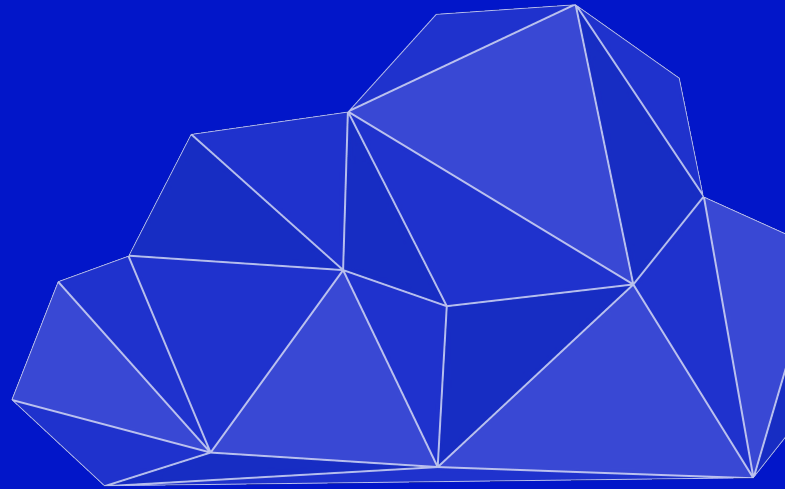
1. Scalability
2. Data availability
3. Security
4. Performance
5. Durability (11 9's)

AWS Storage Solutions

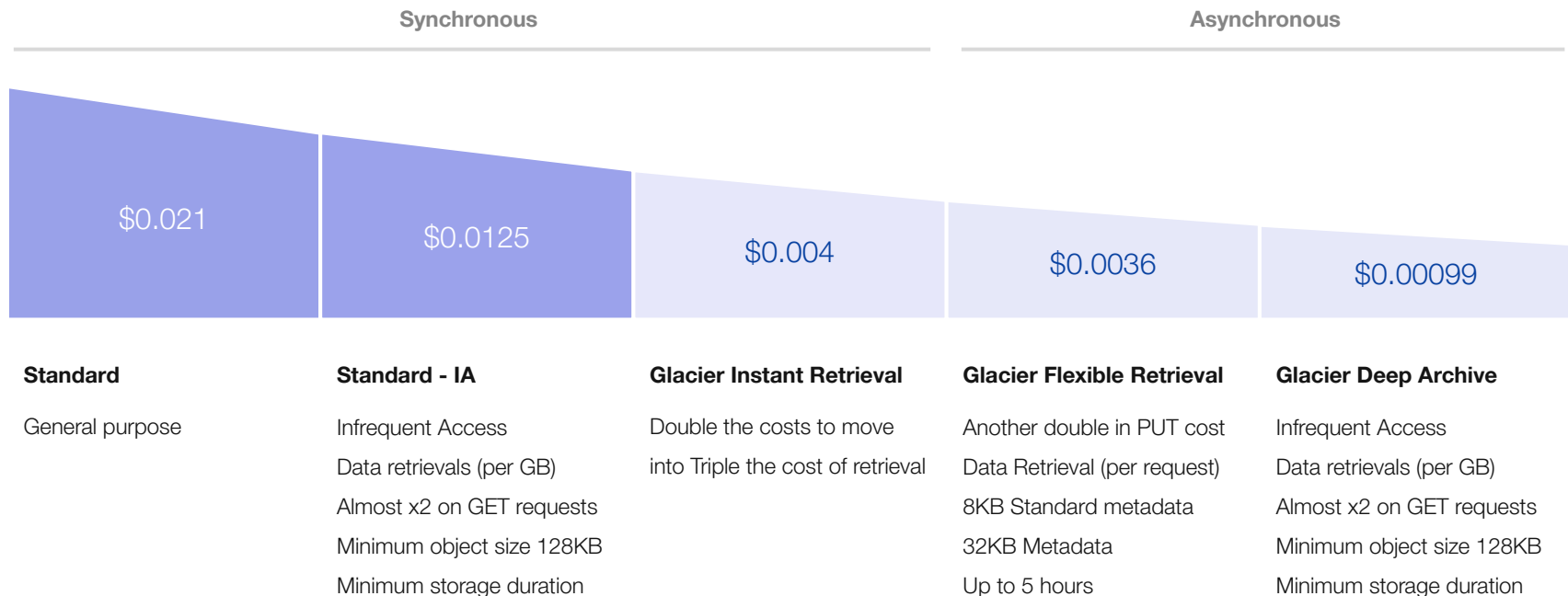


The Challenge

Big Data & Small Letters



Simplified S3 Storage Classes Pricing



The Challenge

**Optimizing multi-dimensional,
dependant cost factors across:**

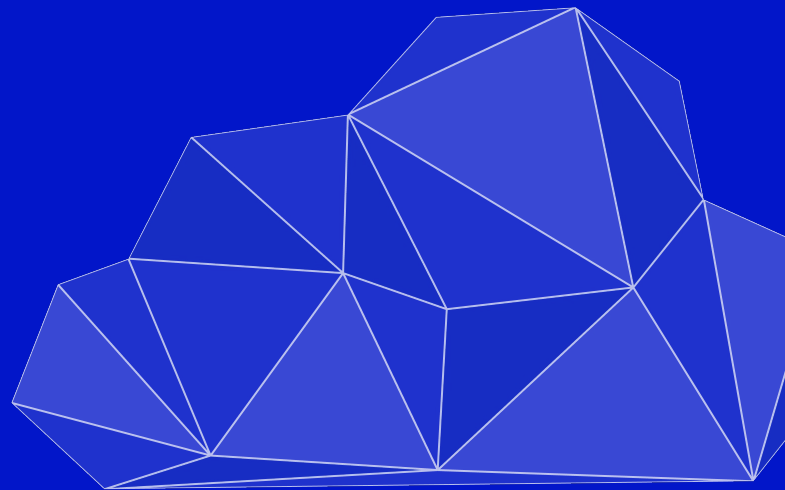
1. Storage
2. Requests
3. Data transfer
4. Transitions
5. Data retrievals
6. Overhead

Not to mention...:

1. Monitoring
2. Encryption
3. Access Point
4. Inventory
5. Access Logs
6. Acceleration
7. Replication
8. ...

The Goal

Efficient Buckets Architecture



An Overlooked Profession: Bucket Architecture

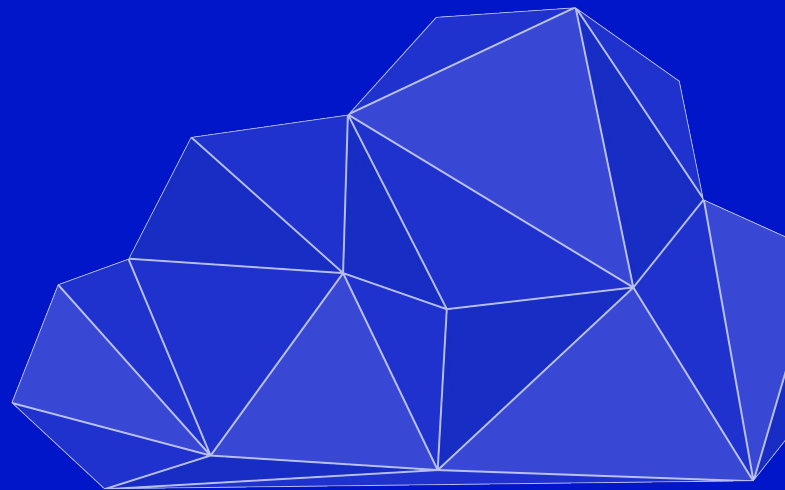
The Ultimate Goal:

Choose the correct storage class for all objects given their usage pattern and attributes

- Buckets are free of charge
- Generalization VS Designation - what are the common attributes?
- Additional concerns:
 - Authorization
 - Data compliance
 - Application architecture
 - ...

The Common (Bad) Practice

Generalized Buckets



The Generalized Bucket - Characteristics

TL;DR: No definitive purpose

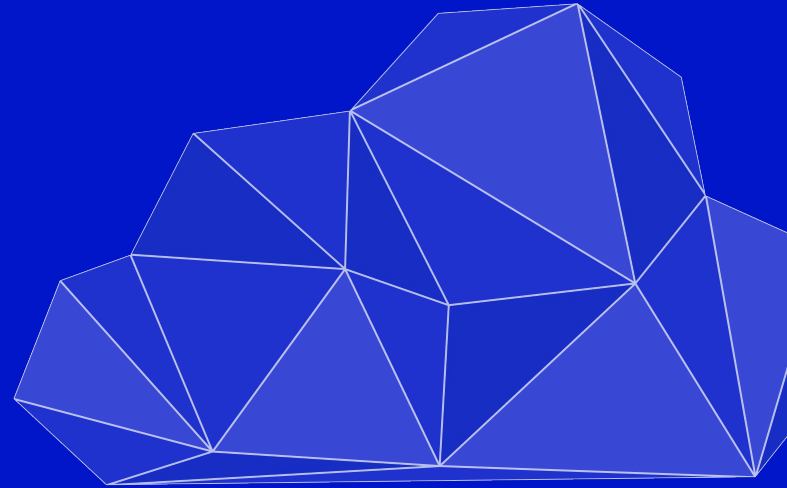
- Reflect an “Org” structure: department, team, project, product
- A mix of content types
- Both hot and cold data
- Multiple (many) writing applications
- Used as hierarchical folders

Examples

Company-level	<i><company>-logs</i>
Region-level	<i><company>-us-east-1</i>
Department-level	<i><company>-<department></i>
Product-level	<i><company>-<product></i>
Service-level	<i><company>-rds-logs</i>

The Ideal

Designated Buckets



The Designated Bucket Utopia - Characteristics

TL;DR: Serves a well defined purpose

- Contains homogeneous content types
- Contains homogeneous object sizes
- Prefixes are used for “indexing” instead of hierarchical folders
- Located near its workloads/clients
- Has minimal number of writers

The Designated Bucket Utopia - Extra Benefits

Better architecture and flexibility with no extra charge, but also:

- Allows best cost allocation because CUR is bucket-level
- Some S3 features are bucket-level and not prefix-level

Examples - AWS Internal Artifacts

AWS CUR

VPC FlowLogs

S3 Inventory

Ever wondered why the configuration of these artifacts allows you (or even directs you) to **create a new bucket**, or at least **define a prefix** for the destination artifacts?

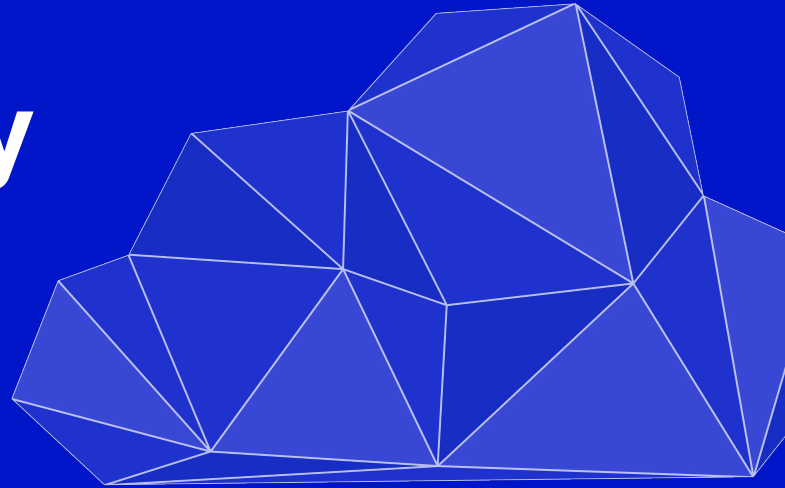
Examples - From the 20th Century

The Linux File System Structure

/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION <small>REQUIRED DIRECTORIES: OPT, X11, SERIAL, XML</small>
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS <small>SECONDARY HIERARCHY</small> <small>REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE</small>
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

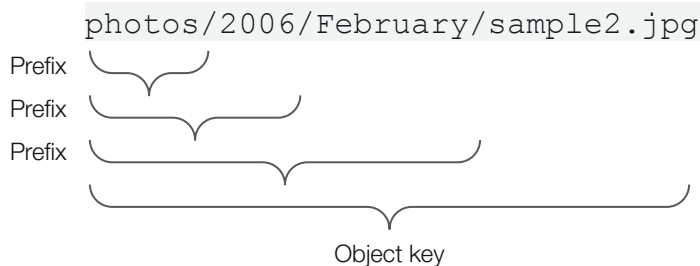
The Methodology

Prefix-Oriented
Objects Management
(POOM)



AWS Official Documentation

*“A **prefix** is a string of characters at the beginning of the object key name. A prefix can be any length, subject to the maximum length of the object key name (1,024 bytes). You can think of **prefixes as a way to organize your data** in a similar way to directories. However, **prefixes are not directories**”*



POOM = Prefix-Oriented Objects Management

Under the hood, prefixes are implicit instructions for S3 to **partition the physical data storage**.

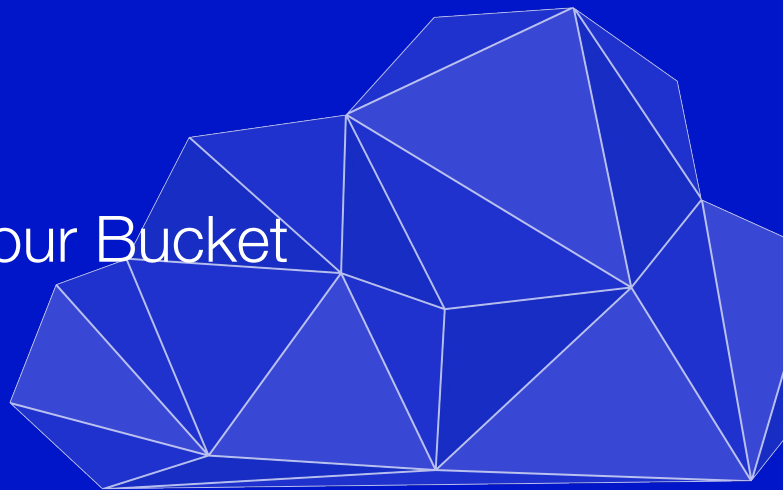
Thus, most relevant S3 mechanisms work by the prefix:

1. Lifecycle Policies
2. Expiration Policies
3. API (prefixes actually let you horizontally scale API requests per second!)
4. Intelligent Tiering
5. Inventory

The Tool



S3DMap: A 3D Visual Map for your Bucket
Storage



<https://github.com/PointFiveLabs/s3dmap>

Which prefixes hold the most storage?

What is the average object size in a prefix?

What file types are there in each prefix?

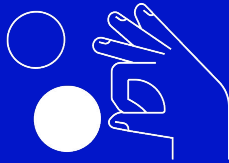
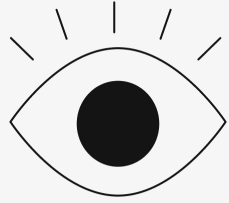
Storage class distribution within a prefix

**How many days since
an object was updated
in a prefix?**

Is this prefix file-type homogeneous?

**How many
non-current versions
per object in a prefix?**

**Which prefixes have
the biggest versioning
cost overhead?**



Thank You



PointFive

