

# Introduction to Data Management

## Data Integrity

Alyssa Pittman

Paul G. Allen School of Computer Science and Engineering  
University of Washington, Seattle

# Announcements

- HW4 due today 11 pm
- Midterm this Monday, 2/10, in class
  - Covers material through Monday's lecture
    - e.g. can expect transactions but not isolation levels
  - Notes: 1 sheet, both sides, handwritten
- TA-led midterm review Sunday 10 am EEB 125

# Data Integrity

What is it?

Data integrity is the maintenance of, and the assurance of the accuracy and consistency of data over its entire life-cycle.

# Motivating Example: Gitlab data loss

Gitlab had a single primary and single secondary database. Due to increased load, replication lagged and started failing.

While responding, the oncall engineer accidentally deleted the live database.

The nightly backups had been failing.

It took over 18 hours just to copy data from a serendipitous recent backup.

Gitlab lost 6 hours of data (estimated at 5000 projects, 5000 comments, and roughly 700 users).

<https://gitlab.com/gitlab-com/www-gitlab-com/blob/master/source/posts/2017-02-10-postmortem-of-database-outage-of-january-31.html.md>

# How much data loss is acceptable?

None?

Is this a realistic guarantee in real-life, distributed systems?

# How much data loss is acceptable?

SLO = Service Level Objective

99% = 2 nines = lose 1 of 100 objects stored

99.999% = 5 nines = lose 1 of 100,000

99.999999999999% = 11 nines = 1 of  $10^{11}$

https://www.hellosign.com/legal/security



PRODUCTS ▾

PRICING ▾

CUSTOMERS ▾

INTEGRATIONS ▾

SOLUTIONS ▾

API ▾

SIGN UP

CONTACT US

LOGIN ▾

Legal

API Terms

Privacy

## RELIABILITY

The system used to store HelloSign documents is designed to achieve 'nine 9s' of durability, with data automatically replicated in multiple data centers.

<https://aws.amazon.com/glacier/>

# Amazon S3 Glacier

Long-term, secure, durable object storage for data archiving

Get started with Amazon S3 Glacier

Request more information

Amazon S3 Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup. It is designed to deliver 99.999999999% durability, and provides comprehensive security and compliance capabilities that can help meet even the most stringent regulatory requirements. Amazon S3 Glacier provides query-in-place functionality, allowing you to run powerful analytics directly on your archive data at rest. Customers can store data for as little as \$0.004 per gigabyte per month, a significant savings compared to on-premises solutions. To keep costs low yet suitable for varying retrieval needs, Amazon S3 Glacier provides three options for access to archives, from a few minutes to several hours.



**Expedited**  
1-5 minutes



**Standard**  
3-5 hours



**Bulk**  
5-12 hours



# Service levels for durability

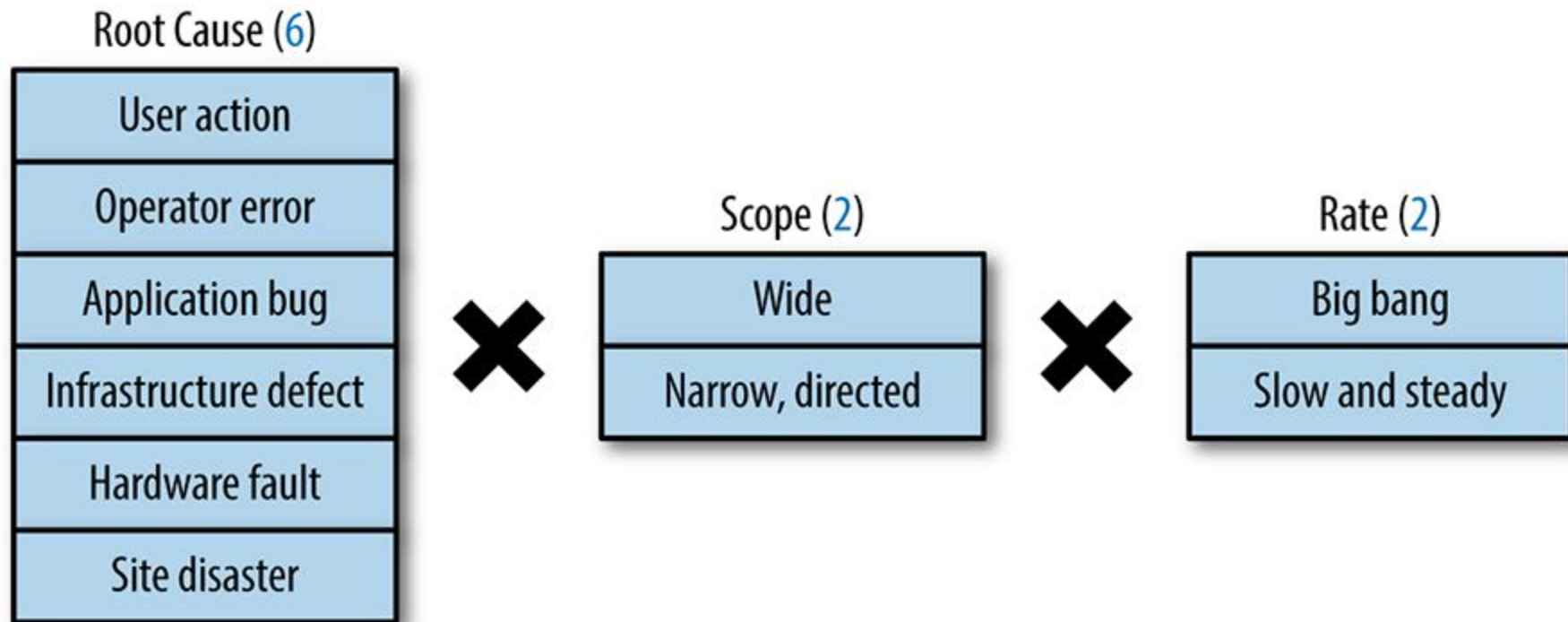
## RPO: Recovery Point Objective

How much data can you lose during a failure?

## RTO: Recovery Time Objective

How long does it take you to get data back online after a failure?

# Types of failures



From [“Site Reliability Engineering”](#), copyright Google Inc , used under [CC BY-NC-ND 4.0](#)

# Replication

Stream changes to extra databases

Solves:

- Data locality

- Failover

But:

- Corruption is quickly replicated too.

# Redundancy

Think of it as distributed RAID

9x3 data storage

All customer data is stored up to nine times across three geographically disparate locations, providing a superior level of data integrity and protection against data corruption.



Solves:

Failover

Restore if some copies are corrupted

From <https://www.docuSign.com/how-it-works/availability>

But:

Corruption/deletion can be propagated on write

# Backups

Periodically backup all data, often in multiple locations/mediums

Solves:

- Recent deletions/corruptions

But:

- Stale backups lose data

- Can be expensive (space, computation)

# Backups

Incremental backups: only save data that's changed since the previous backup

On-demand backups: create a backup when you're about to do a risky update

Transaction logs: log mutations that occurred since the last backup

Storage mediums: tape, off-site, ...

# Backups are useless...

...What people care about is restores.

Monitor backups.

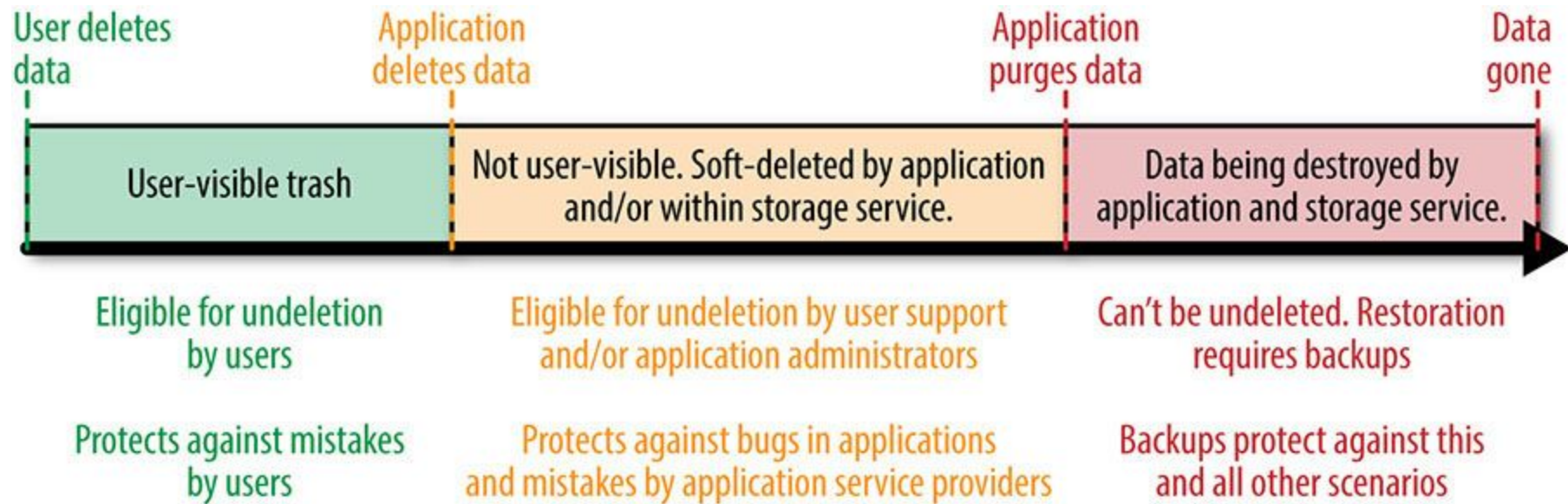
Test your restore process.

Schema changes?

Data migrations?

Point-in-time restore?

# Data retention



From [“Site Reliability Engineering”](#), copyright Google Inc , used under [CC BY-NC-ND 4.0](#)



# Data validation

Can you find (and solve) problems before users notice?

Automated validation pipelines

Checksums

# Don't underestimate the human element

Can you prevent people from easily running commands that destroy data?

...or not have those commands at all?

Can you limit access to the people who really need it?

Can you grant access to enough people that catastrophes (or vacations!) don't cause problems?

# Defense in depth

TL;DR:

Have multiple approaches

And make sure they work!