

The background of the image is a vibrant, abstract composition of overlapping, semi-transparent geometric shapes. These shapes, primarily in shades of blue and purple, create a sense of depth and movement, resembling a stylized architectural structure or a complex data visualization. The lines are sharp and angular, contributing to a modern, tech-oriented aesthetic.

AWS re:Inforce

JULY 26 – 27, 2022 | BOSTON, MA

NIS234

Learn to create end-to-end security invariants using AWS services

Esteban Hernandez (he/him)

Principal Security Solutions Architect
Amazon Web Services

Laurens Brinker (he/him)

Assoc. Security Solutions Architect
Amazon Web Services



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

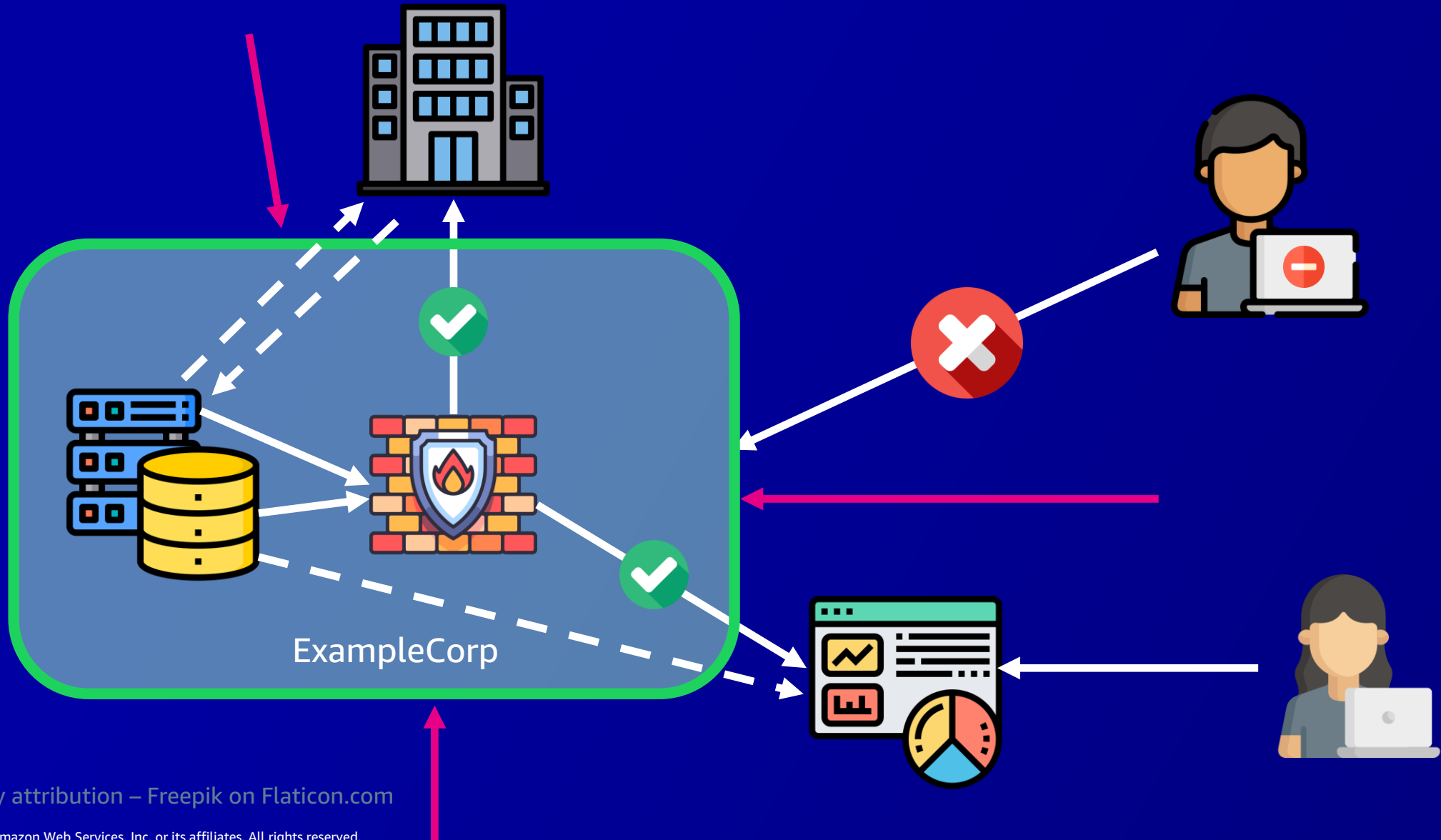
- Setting the scene
- Threat modeling
- Security invariants – including examples with implementation
- Interactive breakout
- Typical challenges with invariants
- Automated reasoning
- Q&A

Note: You're encouraged to ask questions at any point during the session

Security invariants

A **security invariant** is a system property that relates to the system's ability to prevent security issues from happening

Real-life scenario



Icons: License by attribution – Freepik on Flaticon.com

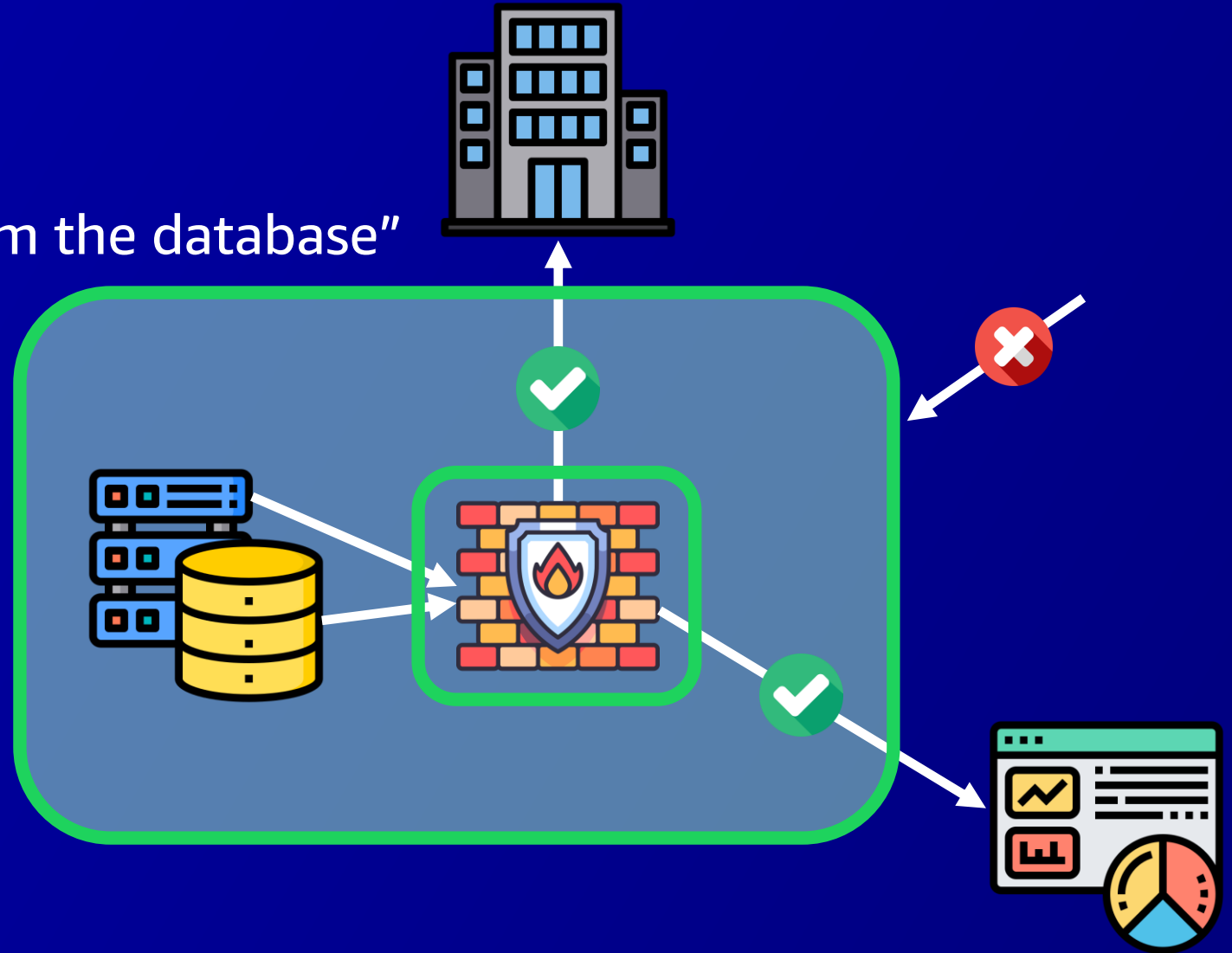


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

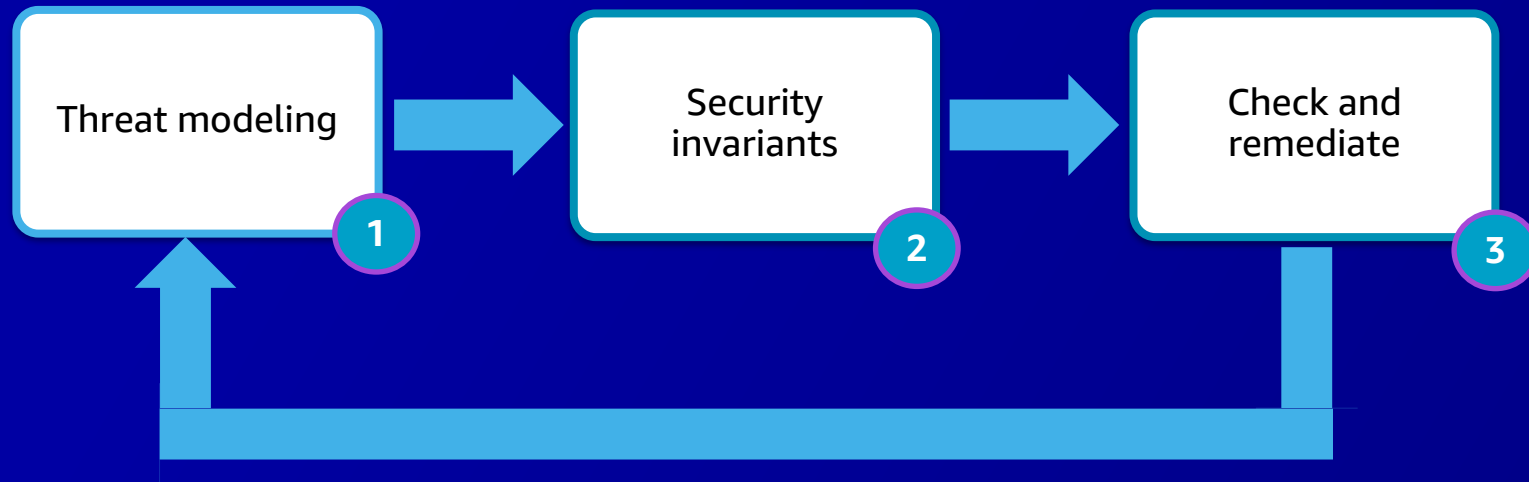
What is the biggest risk you're trying to mitigate?

Any guesses?

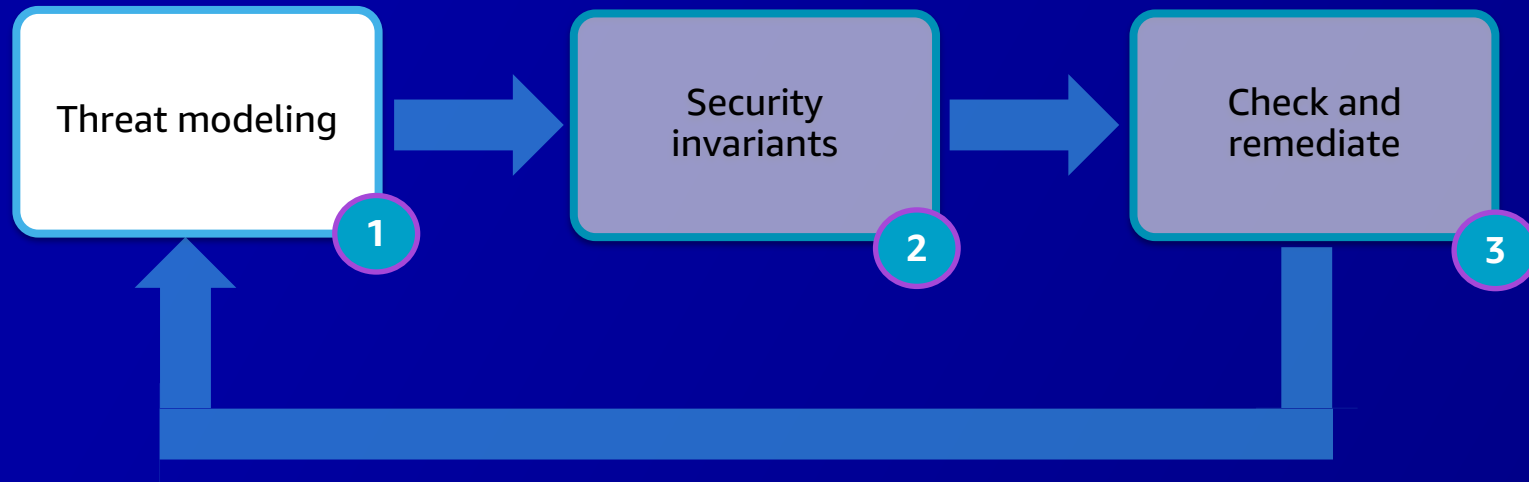
"Exfiltration of sensitive data from the database"



Security invariant life cycle



Security invariant life cycle



Threat modeling

Threat modeling is a mental exercise for identifying threats for the system you are building

A threat model answers four questions*

What are we building?

What can go wrong?

What can we do about it?

Did we do a good enough job (for now)?

* Courtesy of Adam Shostack: <https://github.com/adamshostack/4QuestionFrame>



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Threat modeling – Real-life example

What are we building?

- A system which has sensitive third-party data stored in an enclave
- The system is allowed to fetch additional data (pull)
- It generates a report with aggregated, non-sensitive data for users to consume

What can go wrong?

- Exfiltration of sensitive data from enclave via threat actor

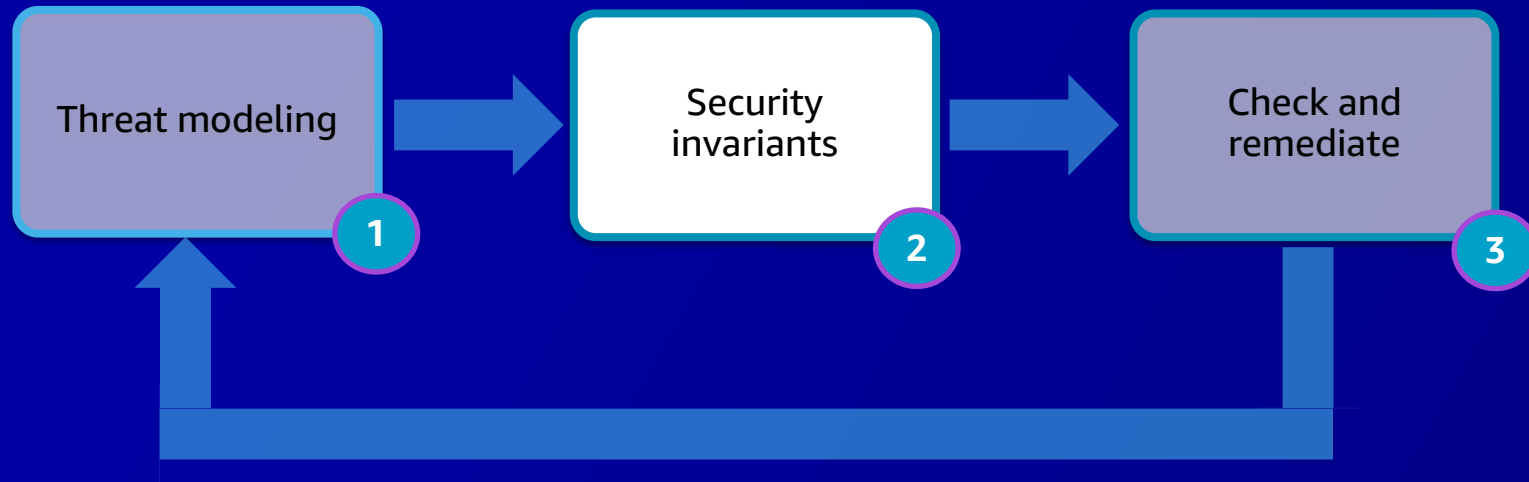
What can we do about it?

- Configure egress filtering to only allow traffic to destinations we trust

Did we do a good enough job (for now)?

- Needs to be tested and revised as changes go through

Security invariant life cycle

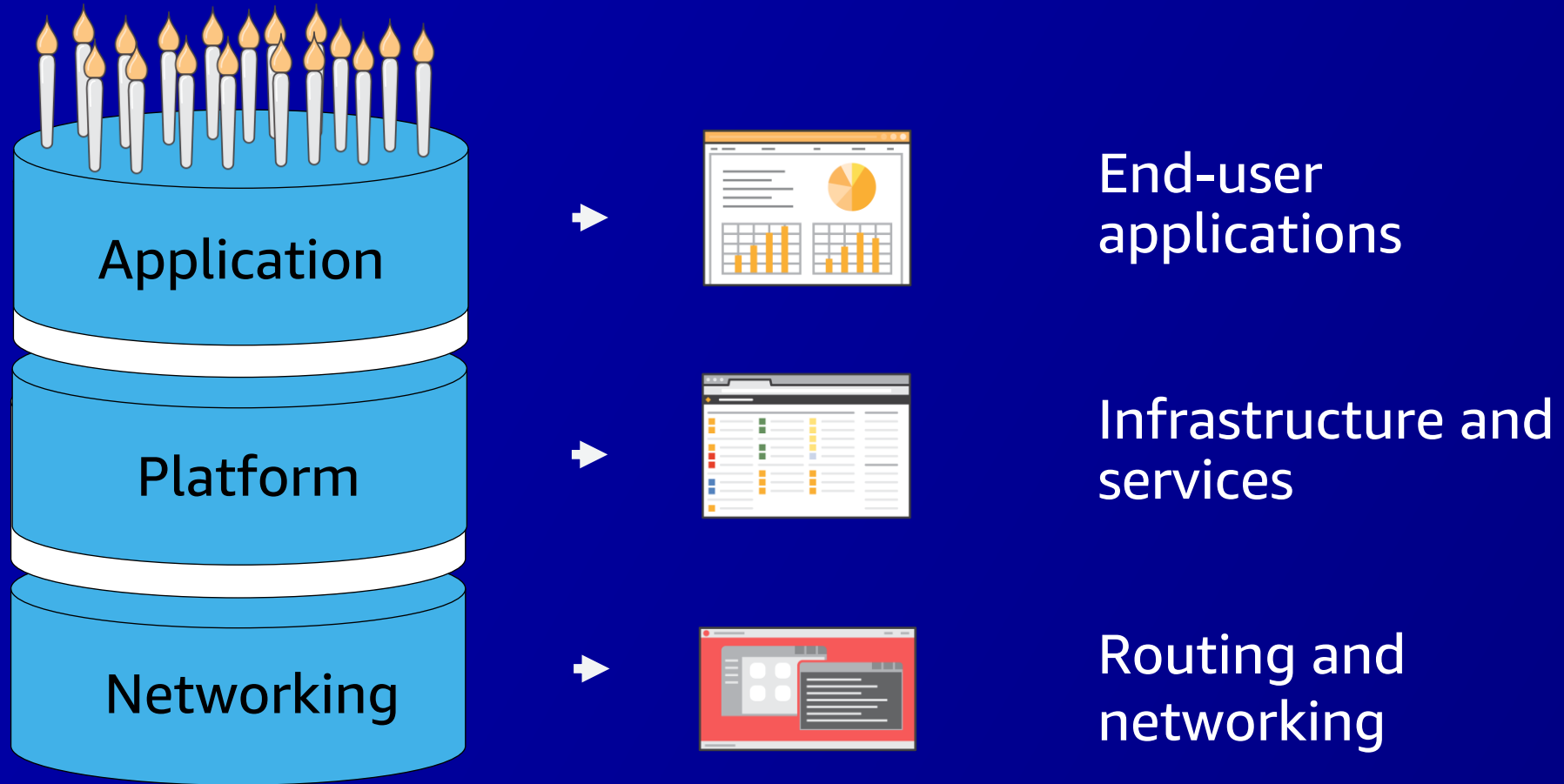


Security invariants

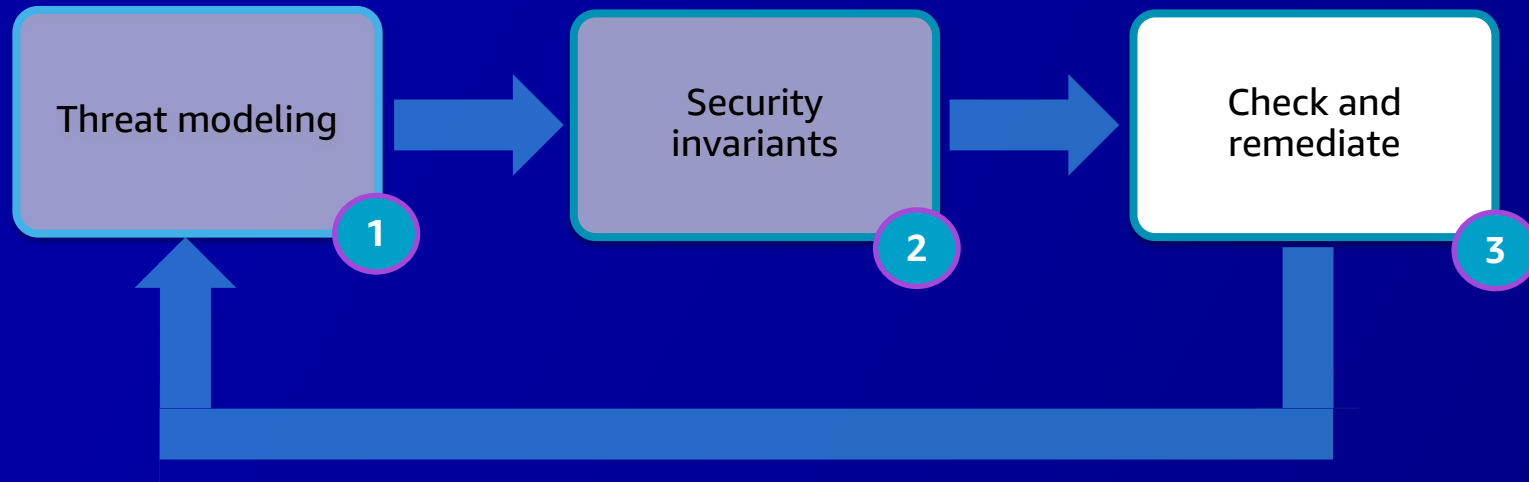
A **security invariant** is a system property that relates to the system's ability to prevent security issues from happening

- Example 1: Egress filtering **only allows traffic to destinations we trust**
- Example 2: To prevent an on-path attacker from intercepting HTTP requests, our **API endpoints are only available over TLS**

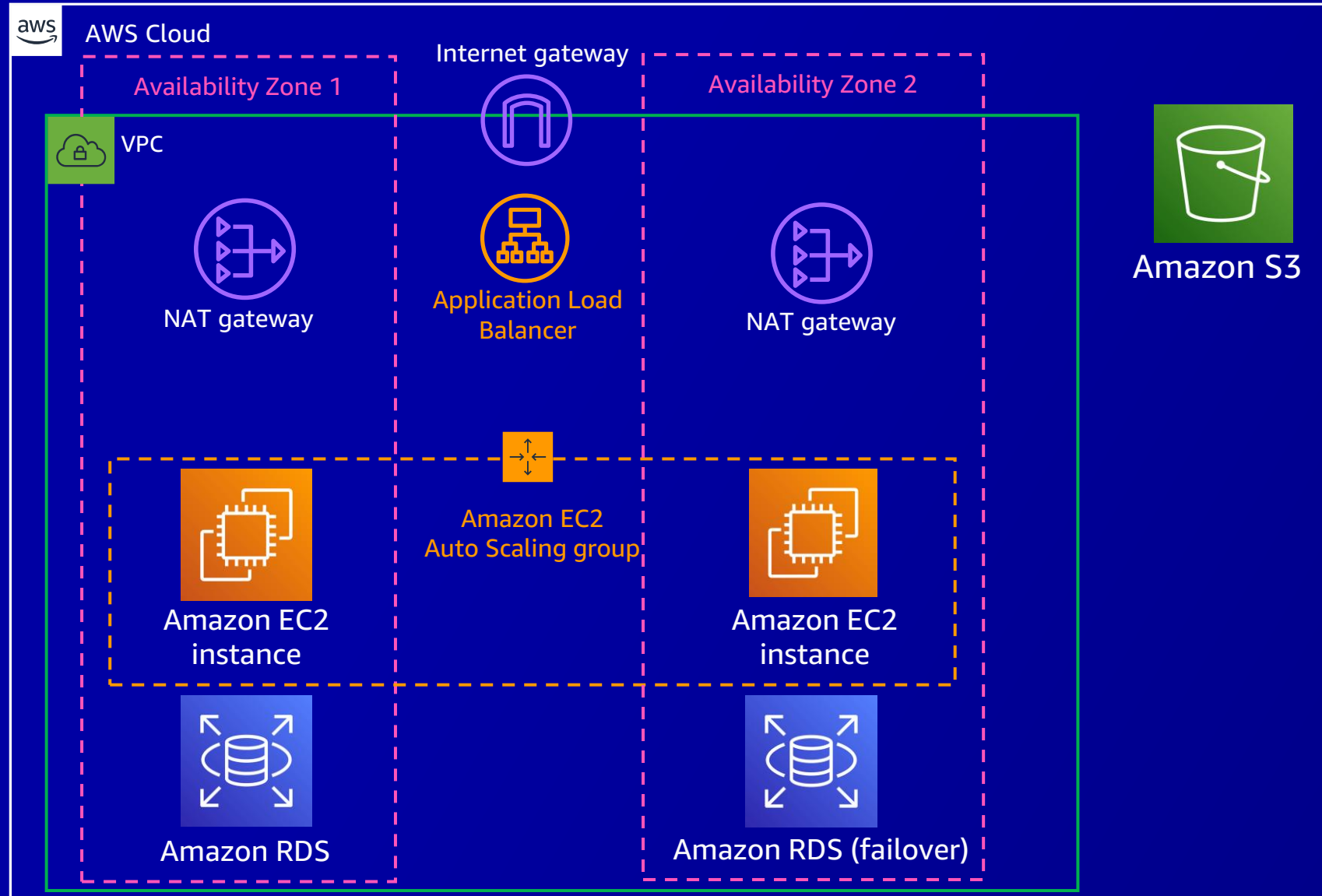
Security invariants – End to end



Security invariant life cycle



Use case – Example 2



Three-tier web application

Allows users to upload and download files via their browser

Security invariant – Networking

Threat: If there is a path from the internet to the database, a threat actor could run queries to extract sensitive information

Invariant: The database is not directly accessible through the public internet

Check: AWS Config Rules (for NACL + security groups) and AWS Firewall Manager

Security invariant – Platform

Threat: The S3 bucket content becomes publicly available to unauthenticated users

Invariant: The S3 bucket does not allow public readability

Check and remediate: Multiple options available – apply block public access at account level and monitor for changes to the configuration of block public access

Security invariant – Application

Threat: Users can access the files of other users if requests are not authorized properly

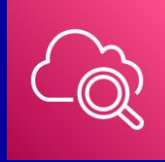
Invariant: Only owners and users with explicit access to a file have access to it

Check and remediate: Run CloudWatch Synthetics canaries. If a canary fails, create ticket and page on-call engineer

Typical AWS services used to check invariants



AWS Config



Amazon CloudWatch
Synthetics



AWS Firewall Manager



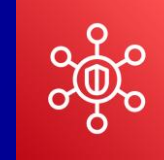
AWS Network
Firewall



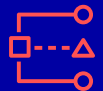
Resolver DNS
Firewall



Service control policies
(SCPs)



AWS Security Hub



AWS IAM
Access Analyzer



Amazon VPC
Network Access Analyzer

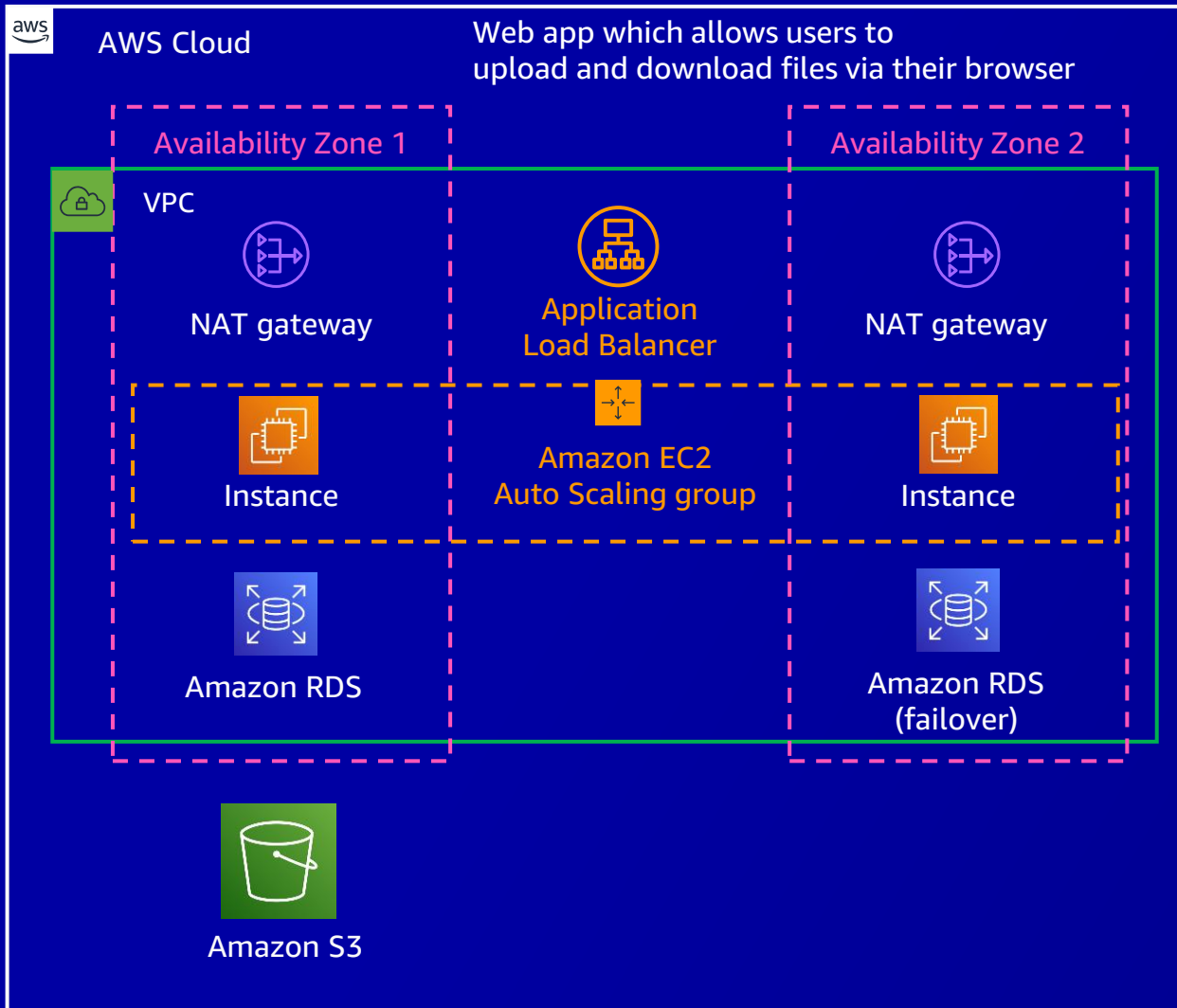


AWS Lambda

Which invariants should I implement?

... it depends

Breakout use case



Step 1: Pick a threat

Bonus points if you can come up with your own threat

Step 2: For the threat, define one or two invariants

Step 3: For each invariant try to come up

- a way to check the invariant
- a remediation action if the invariant deviates from the expected state

Feel free to be creative – add services/functionalities
We will discuss examples together in **10 minutes**

Example list of threats

1. Insufficient input validation could allow cross-site scripting
2. A sustained DDoS attack renders the application unavailable
3. Logs can be tampered with to prevent detection of unauthorized actions

Discussion



Typical challenges

What do you think?

Where/when to execute your invariant checks

Ownership of the invariant

Overlap between invariants

Checking all the edge cases

Provable security

- Mathematical reasoning rather than exhaustive testing
- For example, Network Access Analyzer

Q&A



Related session

TDR204 – Real-world lessons on how to operationalize security findings

Thank you!

