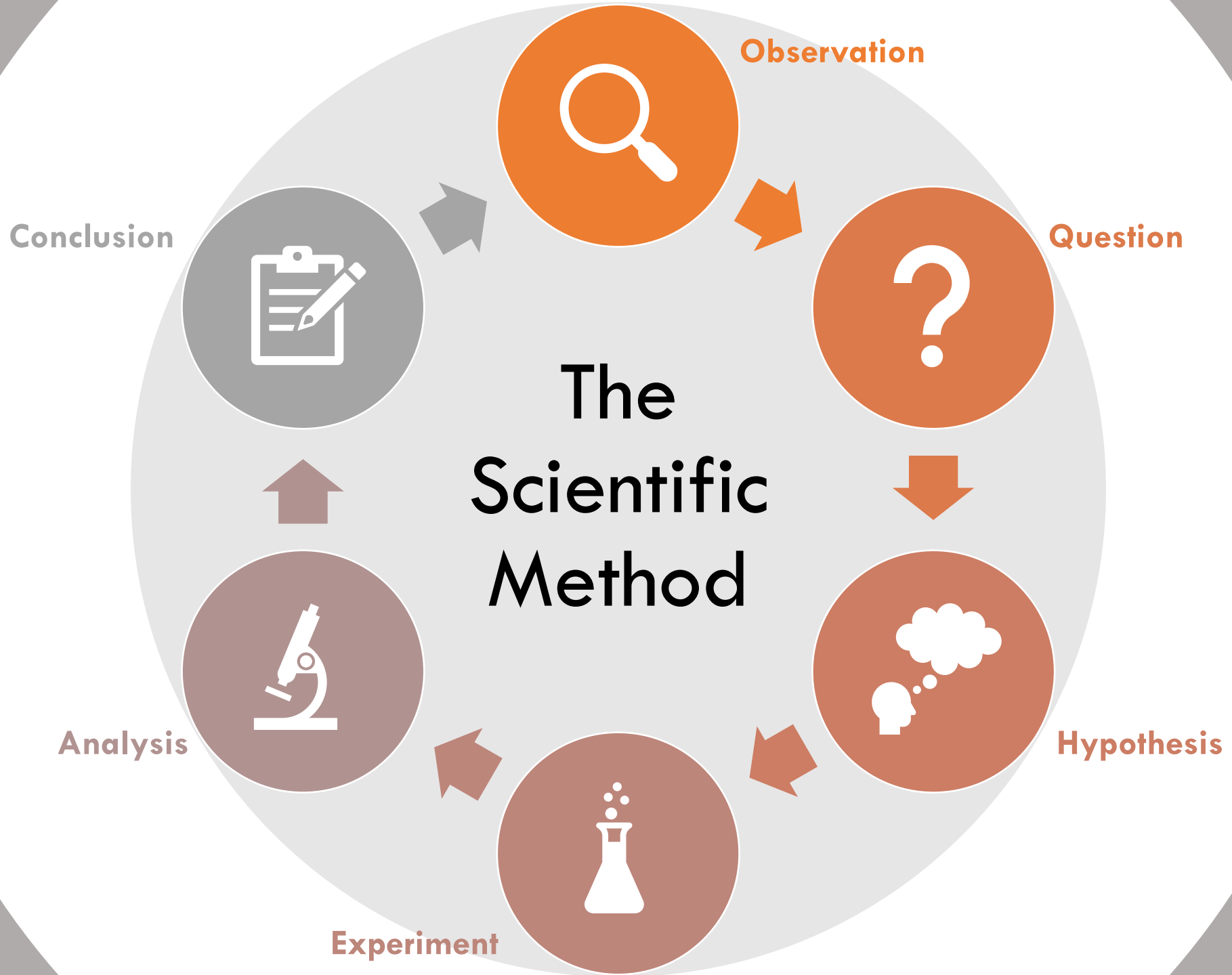


# The Scientific Method for Resilience

Christina Yakomin  
Vanguard



# The Scientific Method



# How does this apply to **resilience?**

1

Failure Modes and Effects Analysis

2

Chaos Engineering

3

Documentation & Planning



# Step 1: Observation

- Reference an architecture diagram
- Identify critical components
- Consider the business process flow

**“Here’s our sample system architecture! Let’s discuss how resilient it is.”**



Client



Web UI



Cloud-based  
Microservice

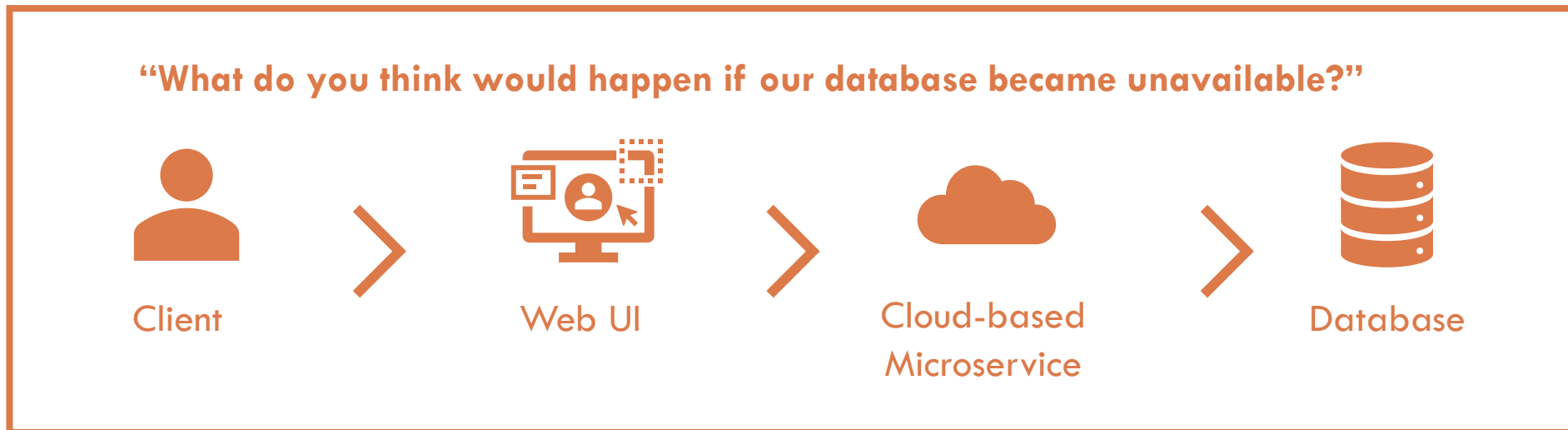


Database



## Step 2: Question

- Discuss how each might component fail
- What would the effect be in each of the failure scenarios?





## Step 3: Hypothesis

- Based on what the team knows about the system, discuss the answers to these questions
- Develop a hypothesis based on the group consensus
- People may not always agree!

**“If our database became unavailable, writes would fail, but reads would be served from our microservice’s in-memory cache.”**



Client



Web UI



Cloud-based  
Microservice



Database



## Step 4: Experiment

- Run a test! Whether you're using a vendor tool, an open source library, homegrown automation, or manual steps – inject the failure mode into the system.

“Let's shut down our database in non-prod to test our assumption!”



Client



Web UI



Cloud-based  
Microservice



Database



## Step 5: Analysis

- Use the available Telemetry/Observability to see the effects of the injected fault
- Compare these observations to the hypotheses. Were the team's expectations met?

“OMG! A retry storm of write requests from our Web UI took out our microservice!!”



Client



Web UI



Cloud-based  
Microservice



Database





## Step 6: Conclusion

- Document your work! Make sure all of the steps are written down and observations have been captured
- Spend some time action planning
- Modify “variables” (make system changes) and repeat!

“Let’s implement better retry logic in our Web UI and adjust our Microservice’s auto-scaling policy so we can be more resilient to database outages. Then we’ll re-test!”



Client



Web UI



Cloud-based  
Microservice



Database

# Christina Yakomin

Senior Technical Specialist  
Site Reliability Engineering at Vanguard

 /in/christina-yakomin

 @SREChristina

 Cloudy with a Chance of Chaos  
SRECon '20

