**Breaking Changes (Chapter 5 – Implementing Microservice Communication):**

* **100 –** This kind of breakage occurs when the structure of an API endpoint changes—such as its path, required parameters, or response format—causing existing clients to fail

**What is a Structural Breakage?**

* **200 –** The calculate method used to add numbers, but now it subtracts them—resulting in which type of breakage?

**What is a Semantic Breakage?**

* **300 –** One of many strategies to avoid breaking changes: add, don’t remove—this is known as what type of change(s)?

**What is an Expansion Change, or Expansion Changes.**

* **400 –** When this number in the semantic version increments, it indicates new functionality has been added in a backward-compatible way.

**What is the Minor Version?**

* **500 (DAILY DOUBLE) –** This strategy for handling breaking changes should be avoided whenever possible, as it compromises independent deployment.

**What is a Lockstep Deployment?**

**Testing 1, 2, 3 (Chapter 9 – Testing):**

* **100 –** Tests designed to help nontechnical stakeholders understand how your system works are known as these types of tests (the top part of Marick’s quadrant).

**What are Business Facing Tests?**

* **200 (DAILY DOUBLE) –** This model helps us think about both the scope and proportion of automated tests at each level.

**What is the Testing Pyramid?**

* **300 –** This can be used to emulate dependent microservices when implementing service tests.

**What is a Mock or Stub?**

* **400 –** When tests become less deterministic, failing intermittently, and making it hard to pinpoint broken functionality, they are known as these.

**What are Flaky Tests?**

* **500 –** I wrote the tests for the stack, and you wrote the implementation to make them pass. What practice is this?

**What is Test Driven Development (TDD).**

**Defense in Depth (Chapter 11 – Security):**

* **100 –** Identify, protect, detect, and respond—along with one more—make up the five core functions of cybersecurity.

**What is Recover?**

* **200 –** the idea that when granting access, we should grant the minimum access a party needs to carry out the required functionality, and only for the time they need it.

**What is Least Privilege or the Principal of Least Privilege?**

* **300 –** give a person (or computer) access to some form of restricted resource (database, computer, user account, or something else).

**What are Credentials?**

* **400 –** Creation, distribution, storage, monitoring, and rotation are all key stages of managing these, or critical pieces of information that a microservice needs to operate and that are also sensitive enough that they require protecting from malicious parties

**What are Secrets?**

* **500 –** When people talk about “HTTPS,” they are typically referring to using HTTP with this protocol.

**What is TLS?**

**Failing Fast (Chapter 12 – Resiliency):**

* **100 –**The ability to absorb expected perturbation.

**What is Robustness?**

* **200 –** This metric is often measured at the 90th percentile to account for outliers in performance data.

**What is Response Time or Latency.**

* **300 –** Can be thought of as an automatic mechanism to seal of a bulkhead in our systems.

**What is a Circuit Breaker?**

* **400 –** This theorem states that a distributed system can only guarantee two of the following three: consistency, availability, and partition tolerance.

**What is the CAP Theorem?**

* **500 –** In these types of operations, the outcome doesn’t change after the first application (even if the operation is applied multiple times).

**What is Idempotent or Idempotency?**

**Mordor (Putting it all together):**

* **100 –** In typical data center speak, we’d talk about this type of traffic as being inside a data center.

**What is “east-west” Traffic?**

* **200 –** When a microservice is upgraded from version 2.0.0 to 3.0.0, what type of changes can be inferred based on Semantic Versioning rules?

**What are Backwards Incompatible Changes?**

* **300 –** Data that is lying around and can become a liability especially if it’s sensitive.

**What is Data at Rest?**

* **400 –** This principle advises against duplicating system behavior and knowledge.

**What is DRY (Don’t Repeat Yourself)?**

* **500 –** occurs when an upstream party tricks an intermediate party into doing things it shouldn’t be doing

**What is the Confused Deputy Problem?**

**Final Jeopardy:**

This testing strategy allows teams to catch breaking changes **before** production, without relying on expensive end-to-end tests. By letting the consumer define expectations, it enables earlier conversations and smarter change management.

