1. Was ist ein "inverse conway manoeuvre"?

* Auflösung der Organisation
* Anpassung der Organisation, um eine bessere Software-Architektur zu erzielen
* Entwickler und Tester sind gleichberechtigt im Daily Scrum
* Auflösung der Architektur-Abteilung

2. Welchen Design Prinzipien folgt eine Microservice Architektur (MSA)?

* Single Responsibility Principle (SRP)
* Minimierung der Verteilung über Server/Netzwerke
* "Best tool for the job"
* Bounded Context als Grundlage der Modularisierung

3. Welche Aussagen über das "Sidecar Pattern" sind wahr?

* Das Sidecar verändert den Source Code der primären Applikation
* Das Sidecar bietet ein homogenes Interface für Plattform Services (z.B. Logging)
* Das Sidecar übernimmt zwingend die persistente Datenspeicherung lokaler Daten
* Das Sidecar läuft neben der primären Applikation (eigener Prozess bzw. Container)

4. Was versteht man unter Conways Gesetz (Conway's Law)?

* Je mehr Microservices desto höher ist die Performance
* Software-Architektur spiegelt die Organisations-Architektur wider
* Organisations-Architektur spiegelt die Software-Architektur wider
* Anzahl der Microservices ist direkt proportional zur Anzahl der Testfälle

5. Warum implementiert man eine Menge von miteinander kommunizierenden Microservices statt einer monolithischen Applikation?

* Höhere Robustheit gegenüber System-Ausfällen
* Reduktion der Kommunikation zwischen einzelnen Services
* Erhöhung der Autonomie einzelner Entwicklungsteams
* Bessere Kapselung einzelner Services

6. Welche Aussagen über Microfrontends sind wahr?

* Microfrontends erweitern das Konzept von Microservices auf die Welt der Frontends
* Das GUI von Micropayment Services wird als Microfrontend bezeichnet
* Den Begriff Microfrontends gibt es seit ca. 2016
* Microfrontends betrachten das GUI als Komposition von Einzel-Features

7. Wie groß soll ein Microservice sein?

* So groß, dass der Code "in den Kopf des Entwicklers passt"
* Größer als ein Monolith
* So groß, dass ein Microservice "in zwei Wochen neu geschrieben werden kann"
* Alle Microservices sollen exakt gleich groß sein (in Lines of Code gemessen)

8. Wenn es im Gesamtsystem n Microservices gibt und alle Microservices untereinander kommunizieren, wie viele Kommunikationspfade ergeben sich dann im Gesamtsystem?

* n über 2
* n über n
* n hoch n
* Keine der angeführten Wahlmöglichkeiten

9. Was sind Vorteile von BASE in Vergleich zu ACID

* BASE Architekturen skalieren besser
* BASE unterstützt die System-Eigenschaft "Isolation"
* BASE ist atomar, aber nur solange notwendig
* BASE unterstützt alle ACID Eigenschaften plus die Integration von Microservices

10. Welche der folgenden Eigenschaften sind Eigenschaften von Monolithischen Systemen?

* Typischerweise hochhochkomplex
* Typischerweise über viele Jahre gewachsen
* Hohes Risiko bei Änderungen, Funktionalität auch außerhalb des Bereiches der Änderung zu beschädigen
* Stets freie Technologieauswahl bei Adaptionen möglich

11. Welche Eigenschaften sind Eigenschaften von monolithischen Softwaresystemen?

* Bestehen aus kleinen und verteilten unabhängigen Services
* Slow time-to-market
* Benötigen Kubernetes als Runtime Plattform
* Vertikale Silo Architektur

12. Was versteht man unter einem "vertical team"?

* Ein Team, das alle Tätigkeiten in der Entwicklung eines Microservices abdeckt
* Ein Team, dessen Teammitglieder auf verschiedene Gebäudeebenen verteilt werden
* Ein Team, das gemeinsam für die Entwicklung eines Microservices verantwortlich ist
* Ein Team, das für die Wartung der zentralen Micro-Datenbank zuständig ist

13. Welche der unten angeführten Methoden sind geläufige Software Design Methoden?

* Big Design Up Front
* Design by Contract
* First Idea of a System
* Design and Succeed (DAS)

14. Welche der folgend genannten Patterns sind keine Microservice Patterns?

* Singleton
* Model View Viewmodel
* Database per Service
* Service per Team

15. Welche der folgenden Eigenschaften sind Eigenschaften einer Microservice Architecture (MSA)?

* ACID-like
* BASE-like
* 1 monolithic application - 1 database
* Evolution by Revolution (i.e. recreation of services)

16. Welche Eigenschaften sind Eigenschaften eines "Modular monolith with decomposed database"?

* Greift auf mehrere Datenbanken zu
* Besteht aus mehreren Modulen
* Greift immer ausschließlich auf eine einzelne Datenbank zu
* Folgt dem Architektur-Stil "Service-Oriented Architecture (SOA)"

17. Was sind Vorteile einer Immutable Infrastructure?

* Stabilere Umgebung durch unveränderliche Bausteine der Infrastruktur
* Ist immun gegen Hacker
* Änderungen werden automatisch detektiert und das System heruntergefahren
* Änderungen erfolgen immer über neue Versionen der Bausteine der Infrastruktur

18. Welche konfigurierbaren Reaktionen bietet ein Circuit Breaker Framework?

* Rückgabe eines Fehlers
* Rückgabe des letzten gültigen Wertes vor Eintritt eines Fehlers
* Einheitliche Zugriffsschnittstelle auf unterschiedliche persistente Datenspeicher
* Rückgabe eines vorkonfigurierten Standardwerts

19. Welche Arten von "Planes" kennt ein Service Mesh?

* Steering Plane
* Pain Plane
* Data Plane
* Control Plane

20. Was sind die Eigenschaften einer Immutable Infrastructure?

* Software Bausteine sind semi-mutable
* Kommunikationsendpunkte müssen vollständig autark sein
* Bausteine der Infrastruktur sind mutable (veränderlich)
* Bausteine der Infrastruktur sind immuable (unveränderlich)

21. Welche grundlegenden GUI-Integrationsansätze verfolgen Microfrontends?

* Server-Side Integration (e.g. using Server Side Includes)
* Integration-less Integration
* Client-Side Integration (e.g. using Custom Elements)
* Shared-Nothing Integration

22. Was sind die Eigenschaften des Microservice Patterns "Transactional Outbox"?

* Das Message Relay kann Events öfters als einmal senden
* Docker Container in der Outbox werden ständig neu gestartet
* Ein Message Relay sendet Events aus der outbox table an den Message Broker
* Als Teil der lokalen Transaktion werden Events in den outbox table eingefügt

23. Was sind Vorteile einer Microservice Architektur (MSA) gegenüber einer monolithischen Architektur (MOA)?

* Alle Microservices laufen auf einem einzigen Server
* Einfacherer schrittweiser Wechsel der Technologie
* Höhere Wandelbarkeit der Software
* Einfachere horizontale Skalierbarkeit

24. Wann würden Sie Ihrer Chefin die Verwendung von Microservices vorschlagen?

* Einzelne Microservices können rasch vollständig neu implementiert werden
* Microservices unterstützen einfache horizontale Skalierung
* Microservices unterstützen eine hohe Evolvierbarkeit (=Evolutionsfähigkeit) der Software
* Für jedes Service kann die geeignetste Technologie zur Anwendung kommen

25. Welche Features werden über ein Service Mesh typischerweise angeboten?

* Security (Sicherheit)
* Reliability (Zuverlässigkeit)
* Observability (Beobachtbarkeit)
* Accounting of service usage (Verrechenbarkeit der Servicenutzung)

26. Welche Formen von Coupling gibt es?

* Deployment Coupling
* Implementation Coupling
* Domain Coupling
* Temporal Coupling

27. Was ist das Sidecar Pattern?

* läuft in einem eigenen Prozess
* ändert den Applikationscode
* sorgt für Datenpersistierung
* wird z.B. bei Service Meshes eingesetzt

28. Was sind Key-Use Cases bei Service Mesh?

* Secure service communication
* Access control
* Service discovery
* Traffic governance
* Operation reliability

29. Was sind die "Core-Ideas" von Microfrontends

* Teamcode isolieren
* Resiliente Seiten bauen
* Be technology agnostic
* Establish Team Prefixes
* Browser features over Custom API

30. Was bedeutet Canary Testing?

* Nur ein Teil wird zur neuen Funktionalität weitergeleitet
* Soll dazu helfen, keinen Big Bang durchführen zu müssen
* irgednwas was falsch ist