**Self-check questions:**

* What are the cons and pros of the Monolith architectural style?

Pros:

1. Easy deployment
2. Simplicity
3. Low Latency
4. Simplified data Management
5. Cost-Effective
6. Easier-Debugging
7. Simplified testing

Cons:

1. Limited Scalability
2. Limited Technical Flexibility
3. Longer Deployment Cycle
4. Legacy Technologies
5. Lack of Flexibility
6. Reliability/Maintenance (isolate and fix bugs)

* What are the cons and pros of the Microservices architectural style?

Pros:

1. Scalability Improvements
2. Technology Difference
3. Faster development and Deployment
4. Improved Fault Isolation (If one service fails, much less likely others will fail)
5. CI/CD (microservices are well suited for automated testing, deployment for individual service)

Cons:

1. Complexity (designing, developing, and managing)
2. Testing (due to their independencies, E2E testing can become difficult)
3. Monitoring and Debugging
4. Resource Overhead (Each service consumes resources, which can lead to increased infrastructure costs)

* What is the difference between SOA and Microservices?

Key difference is **scope** and coupling of **components**.

**SOA =>** enterprise-scope (defines rules, APIS, communication protocols, business rules for multiple services)  
SOA focuses on providing a unified, strategic view of how services are designed, deployed, and integrated across the entire enterprise.

**Microservices =>** application-scope – meaning the service(element) is specified to a particular application and it has meaning only written the boundaries of that app. It does not define how applications talk to one another- for what we back to enterprise scope.  
Microservices are designed to address the architecture and development concerns of a single application or service. They are highly focused on the specific functionality and requirements of that application.

* What does hybrid architectural style mean? Think of your current and previous projects and try to describe which architectural styles they most likely followed.

**Hybrid**: refers to an approach that combines elements or principles from two or more diff. architectural styles.

(ex. Microservices and SOA , serverless and Container Based)  
  
I am working on project that is legacy code and has Monolithic architecture style. For now, we are working on migration to Azure and planning to change the architectural style in next year.

I was lucky to work also on project designed as Microservice arch. style. My team was responsible for the services designed as FAAS (used Azure functions) which was designed as HTTP trigger based and invokes a function with an HTTP request.

* Name several examples of the distributed architectures. What do ACID and BASE terms mean.

Example of distributed arch. Styles => Layered/N-Tirer, Client-Server, Microservices, SOA.  
ACID is commonly associated with traditional relational databases, while BASE is often used in NoSQL and distributed database systems

**ACID**:

**A**- **Atomicity (**Transactions are treated as single, indivisible units of work**)  
 C- Consistency (**A transaction takes a database from one consistent state to another)

**I- Isolation (**Transactions are isolated from each other**)**

**D– Durability (**Once a transaction is committed, its effects are permanent**)**

**BASE**:

BA- **Basically Available**: There will be response to every request, even if response is system failure or no data to return.

S-**Soft state:**  Data may not be consistent in real time.

E-**Eventually consistent:** Ensures that the system will reach a consistent state eventually.

* Name several use cases where Serverless architecture would be beneficial.

1. Trigger based tasks / Event-driven Processing.
2. Security checks
3. Testing and Development (CI/ CD – code commit triggers to create build)
4. API gateway (can route incoming request to appropriate microservice, manage auth.)