Service Oriented Architecture

By James Brown

06/14/2020

TOPICS

- What service-oriented architectures are.
- What enterprise service buses are.
- Relationship between an ESB and SOA architecture.
- How data is transmitted through a SOA environment.
- The advantages of a SOA architecture.
- The disadvantages of a SOA architecture.
- How software in a SOA architecture is deployed and managed in a production environment.
- How to scale a SOA environment.

What are service-oriented architectures?

- What service-oriented architectures are.
- -SOA is an architecture wherein numerous applications or (services) can be combined into a larger application.
- SOAs are made up of services, data store, monitoring, and ESB. (Krasso)
- Makes applications interoperable
- Applications are grouped logically.
- All applications are able to communicate with each other through an enterprise service bus.
- Can be integrated with services that use SOAP/REST

What is an enterprise service bus?

- ESB's main responsibility is to route traffic to services, data store, monitoring.
- The ESB makes the connection to the applications that are integrated
- Makes SOA based applications easy to scale as new applications can be added as long as they adhere to the ESB pre-defined rules.
- Supports XML/JSON

Relationship between an ESB and SOA architecture.

- ESB is how applications are able to talk to each other, whereas SOA is the overall architecture design.
- SOA splits each application out logically and the ESB is how the consumers interact with the various applications.

How data is transmitted through a SOA environment

- Services transfer data through ESBs.
- ESB's provide flexibility in data types that can be transmitted
- Data is transmitted through requests/responses
- All services connect to one centralized location, allowing communication across the entire organization.

The advantages of a SOA architecture

- Makes large applications easier to scale as services are not tightly coupled
- Makes applications re-usable. With SOA, you can use a single service in as many applications as needed. This reduces redundancies.
- Reliability – As the services are split out, each service is self-contained.
- Monitoring – provides logging and insight into how the application is running.
- Allows you the ability to integrate an existing application into a service.

The disadvantages of a SOA architecture.

- Initial investment can be expensive.
- ESB is the only way for the services to talk to each other. This can be a disadvantage if the ESB has issues, which means no back-ups if the bus goes down.
- Trust can become an issue as each service may be developed by another team.
- SOA has a lot of overhead on hardware as each request has to be filtered.

How software in a SOA architecture is deployed and managed in a production environment.

- A business goal is defined
- The Service to meet the business goal is defined
- Service is created
- Service is mapped to the existing infrastructure
- Service is integrated in a staging environment
- The newly onboarded service is monitored and updated if changes are needed
- Once testing is complete, the service is pushed to production and monitored.
- If changes are needed later, they are first done so in a testing environment, then rolled out to production.

How to scale a SOA environment

- SOA protocols and requirements are defined
- Each service that needs to be added must follow these pre-defined protocols
- By maintaining requirements, it keeps things consistent and allows scalability to happen easier.
- ESB must be stateless, so that it can be duplicated if needed.
- Scaling will probably not be a one-size fits all approach, as each business is a little different.

Sources

- Krasso, P. (2020). Retrieved June 14, 2020, from slack
- Richardson, L., & Amundsen, M. (2013). *RESTful Web APIs.*

Thank you for your time.
Questions?