Kalix

Serverless Computing - WOSCX 16.6.2022 Alan Klikic



Kalix is the <u>PaaS</u> that enables <u>any</u> developer to easily build large-scale, high performance microservices with no operations required







Managed by Cloud / Service Provider







Managed runtime infrastructure



Container-based deployments

CI/CD integrations



Logging & Monitoring



Auto-Scaling



Security







Stateful data handling



Communication patterns



Event Sourcing and CQRS patterns







KALIX





```
//API
message AddItemRequest {
   string cart_id = 1;
   string item_id = 2;
   int32 item_quantity = 3;
}
message CheckOutRequest {
   string cart_id = 1;
}
//domain
message Cart {
   string cart_id = 1;
   repeated CartItem items = 2;
}
```

```
message CartItem {
   string item_id = 1;
   int32 quantity = 2;
```

//service

service ShoppingCartService {
 rpc AddItem (AddItemRequest) returns (Cart);
 rpc CheckOut (CheckOutRequest) returns (Cart);



- API first development model
- Design API endpoints
 - Just configure required communication patterns (HTTP, gRPC, RPC, streaming, eventing)
- Design persistence domain
- Focus on data structure, not database



• • •

```
//service
service ShoppingCartService {
    option (kalix.codegen) = {
        event_sourced_entity: {...}
    };
    rpc AddItem (AddItemRequest) returns (Cart);
    rpc CheckOut (CheckOutRequest) returns (Cart);
}
```



Choose a durable store model

- Key/Value, Event Sourcing, CRDTs
- Simple model change by annotation



•••

public Effect<Empty> addItem(Cart currentState, AddLineItem addLineItem) {
 //validation against current state
 if(currentState.isCheckedOut()) {
 return effects().error("Cart already checked out.");
 l

//state model: event sourcing

ShoppingCartDomain.ItemAdded event = ...
return effects()
 .emitEvent(event)
 .thenReply(newState -> Empty.getDefaultInstance());

```
//state model: key/value
return effects()
    .updateState(newState)
    .thenReply(Empty.getDefaultInstance());
```



Write business logic

- Focus on business logic implement function
- Data (state) automatically injected in the function – no DB access



Under the Hood





Under the Hood



Every proxy is an Akka node...



Future of Serverless Computing

- How do you see Serverless Computing in five years ?
 - DX for new wave of cloud and edge native apps and use-cases
- Propose a technical challenge to solve in this field in the next years
 - new architecture for the edge with more fine-grained data replication, closer to the end user, physical co-location of data, processing, and user
 - compose cloud and edge in a single coherent whole
- What question would you want to ask another participant?
 - How can we tackle security compliances in serverless environments?