



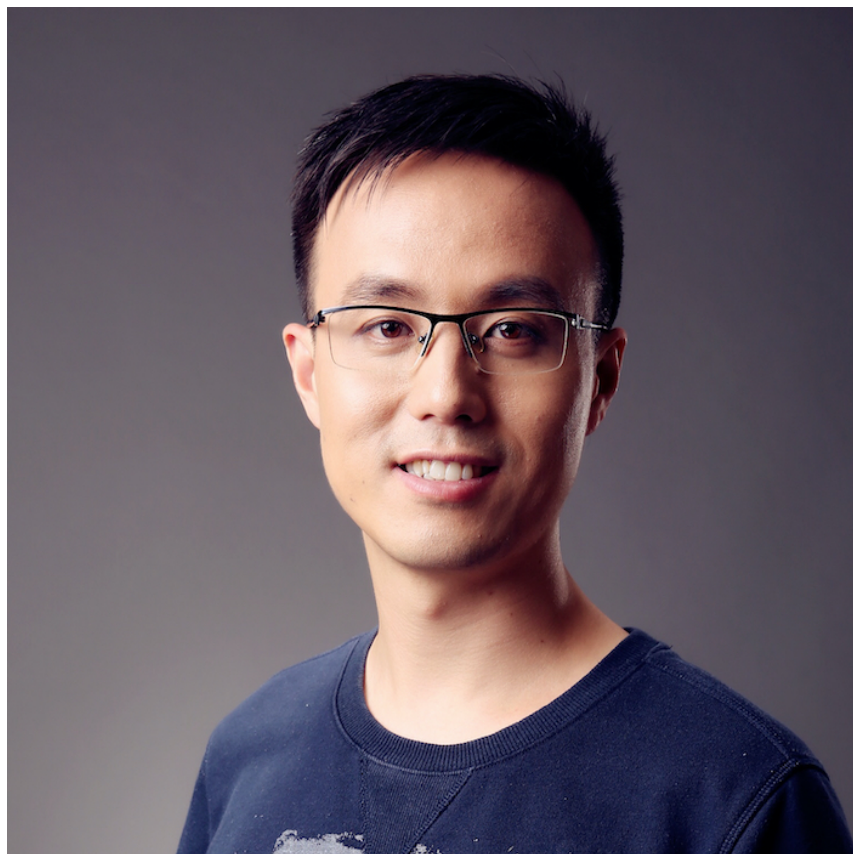
CLOUD NATIVE
Wasm DAY
NORTH AMERICA

WebAssembly + Application Runtime = A New Era of FaaS?

Jason Song



CLOUD NATIVE
Wasm DAY
NORTH AMERICA



Staff Engineer @ Ant Group
Focus on cloud-native and microservice areas, such as Service Mesh, Serverless, Application Runtime, etc.



Open source enthusiast, PMC member of Apollo Config
<https://github.com/nobodyiam>

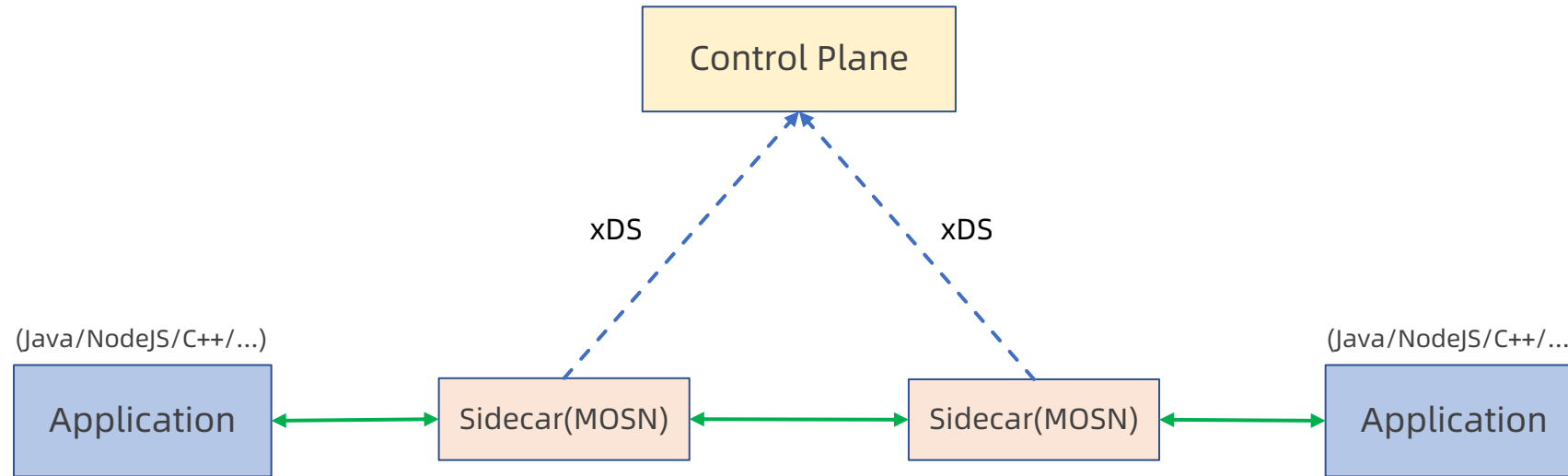


Worked at Dianping and Ctrip, in charge of backend systems and middleware products

Service Mesh Adoption Status in Ant Group



CLOUD NATIVE
Wasm DAY
NORTH AMERICA

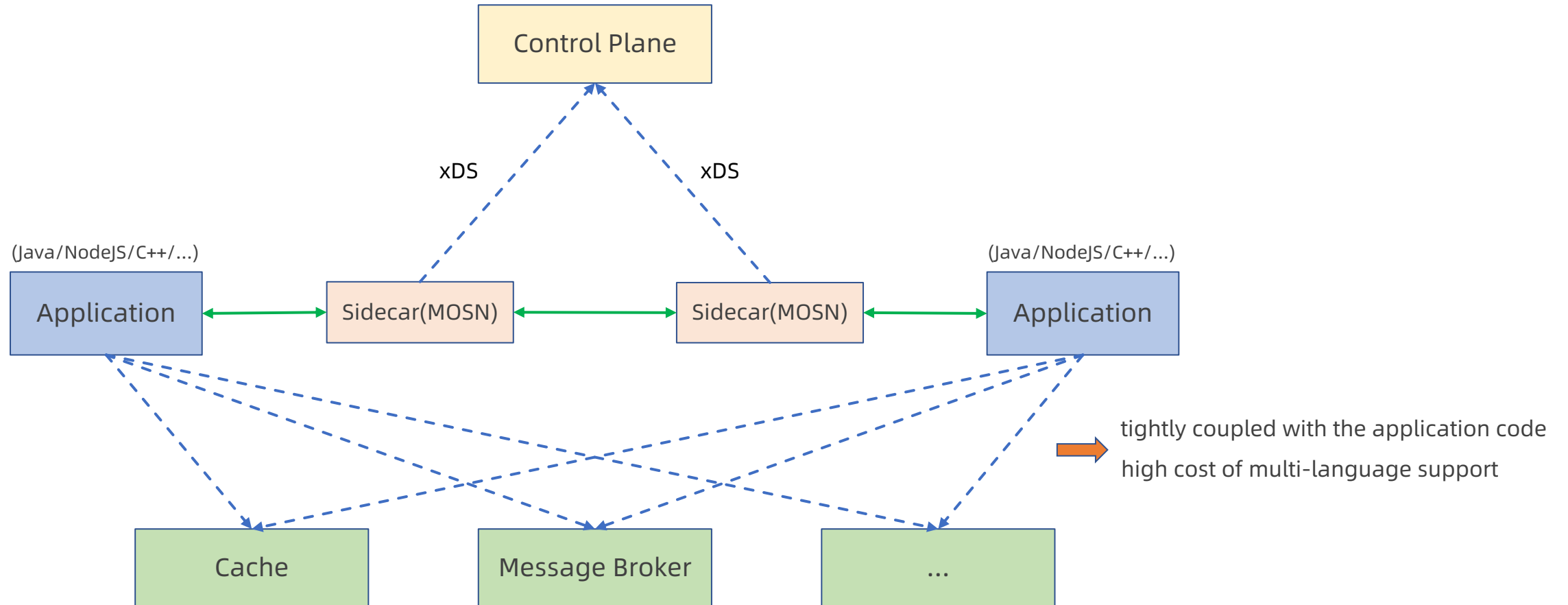


<https://github.com/mosn/mosn>

Challenges remain...



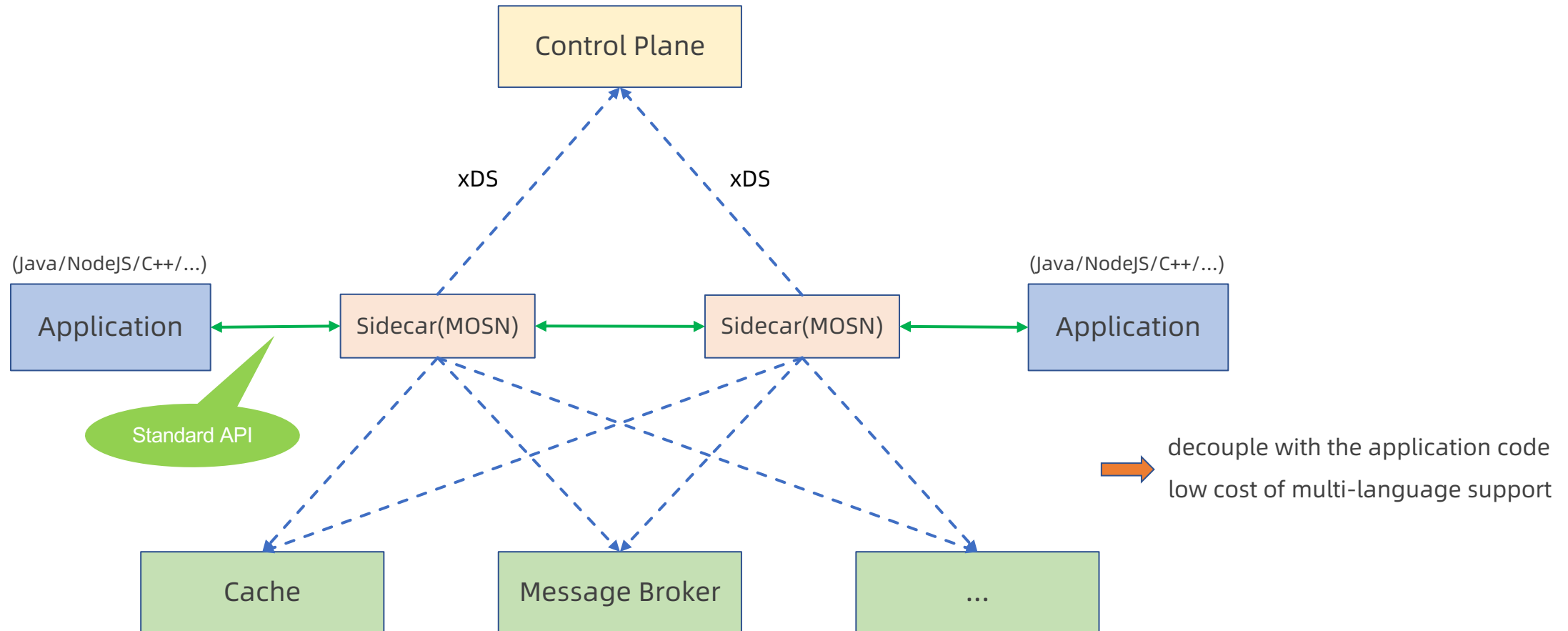
CLOUD NATIVE
Wasm DAY
NORTH AMERICA



Introduction to Application Runtime



CLOUD NATIVE
Wasm DAY
NORTH AMERICA

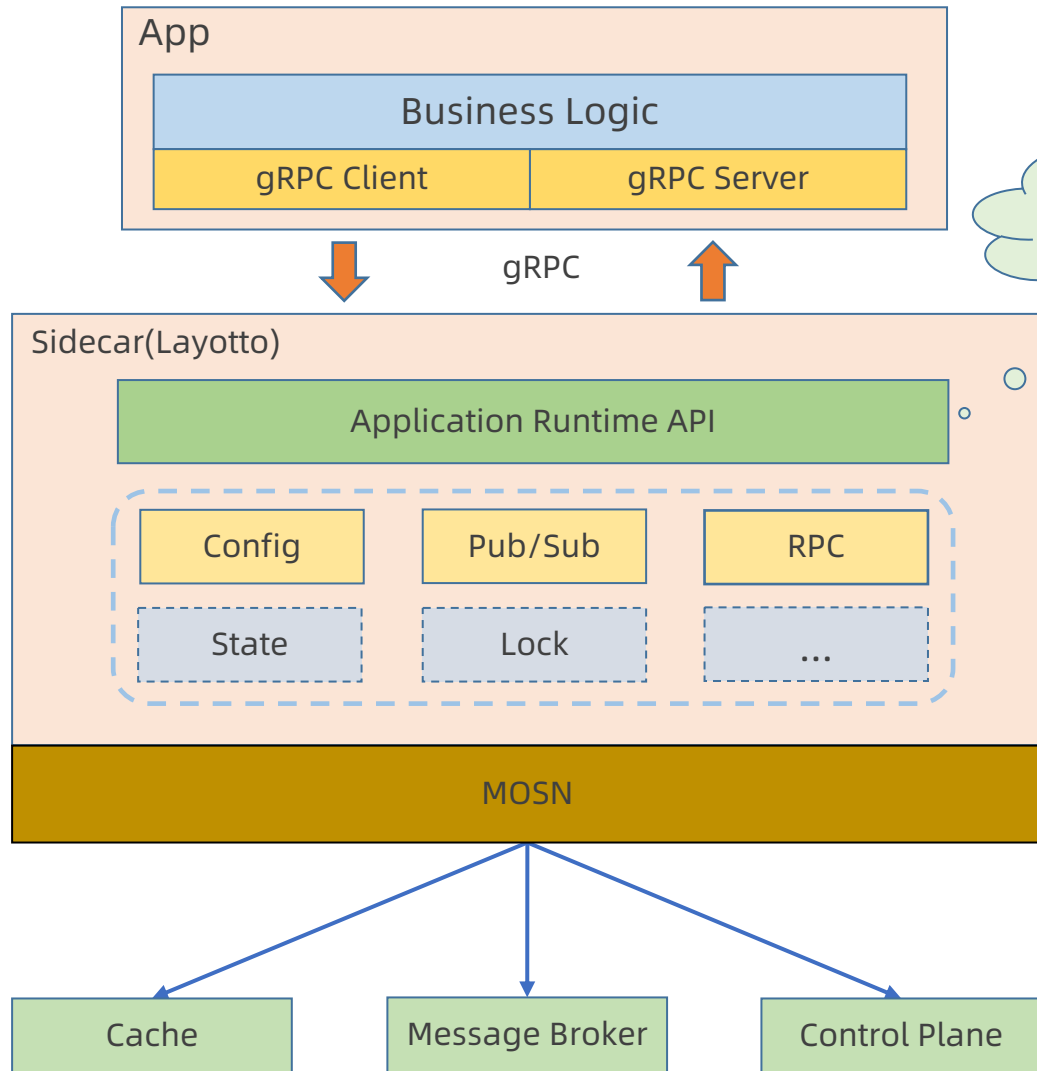


Reference: <https://www.infoq.com/articles/multi-runtime-microservice-architecture/>

Introduction to Layotto



CLOUD NATIVE
Wasm DAY
NORTH AMERICA



Co-build with dapr
community and
alibaba

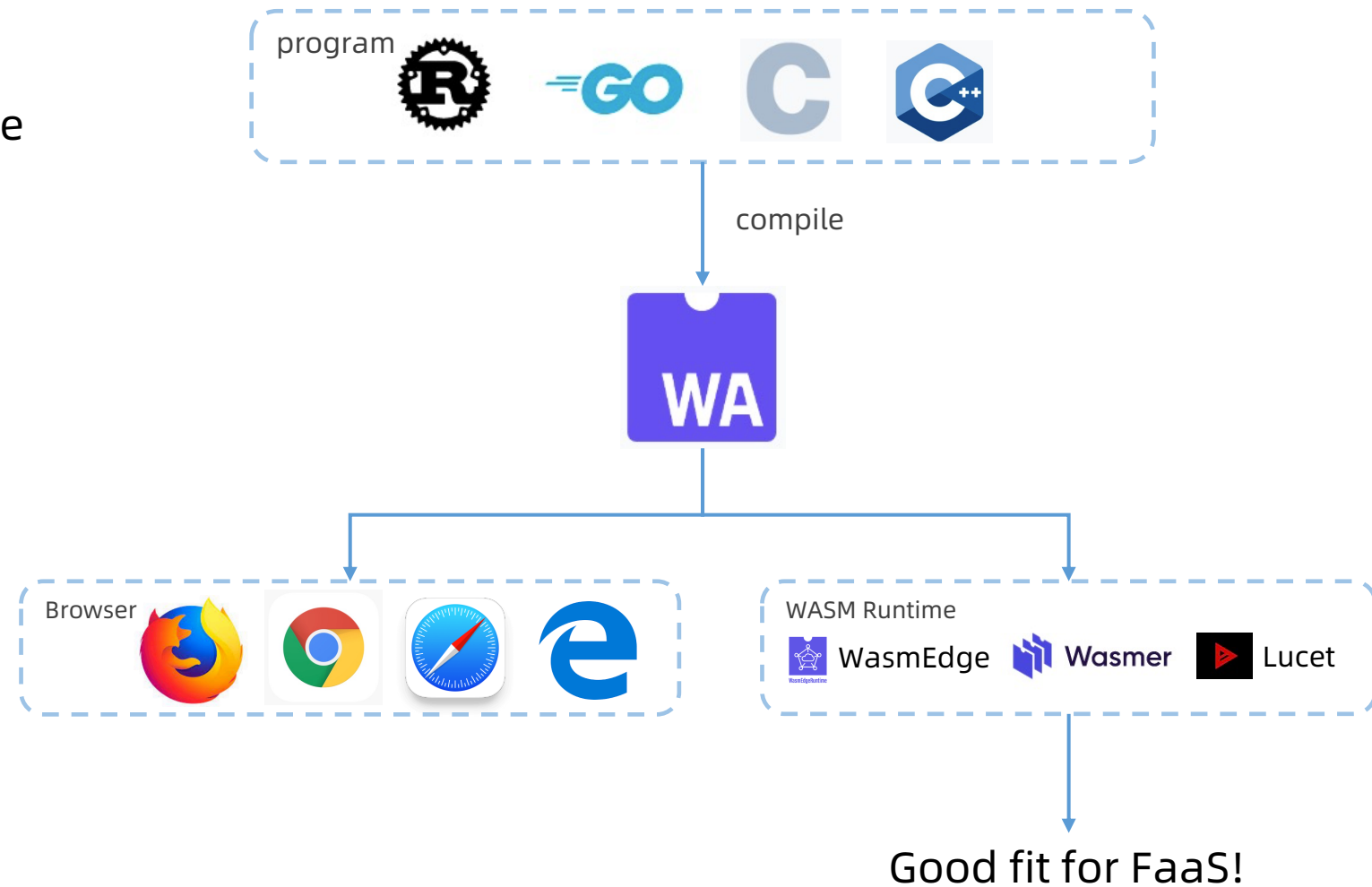
```
service Runtime {  
  // InvokeService do rpc calls  
  rpc InvokeService(InvokeServiceRequest) returns (InvokeResponse) {}  
  
  // Gets the state for a specific key.  
  rpc GetState(GetStateRequest) returns (GetStateResponse) {}  
  
  // Saves an array of state objects  
  rpc SaveState(SaveStateRequest) returns (google.protobuf.Empty) {}  
  
  // Publishes events to the specific topic.  
  rpc PublishEvent(PublishEventRequest) returns (google.protobuf.Empty) {}  
  
  ...  
}
```

FaaS Exploration



CLOUD NATIVE
Wasm DAY
NORTH AMERICA

- near-native performance
- sandboxing
- portability





System Resources, e.g. files, sockets

Invoke services?

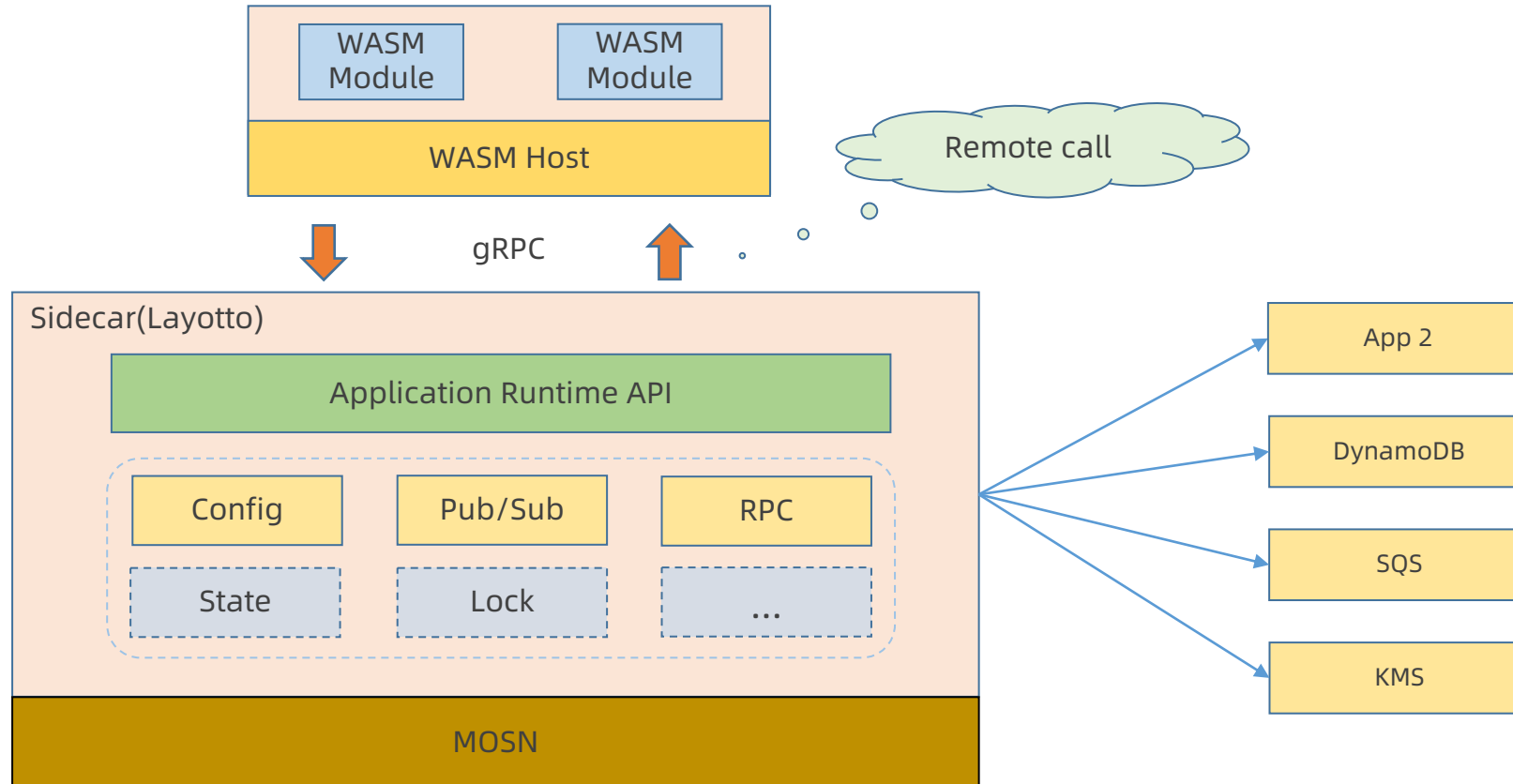
Get/set cache?

Produce/consume messages?

FaaS Exploration



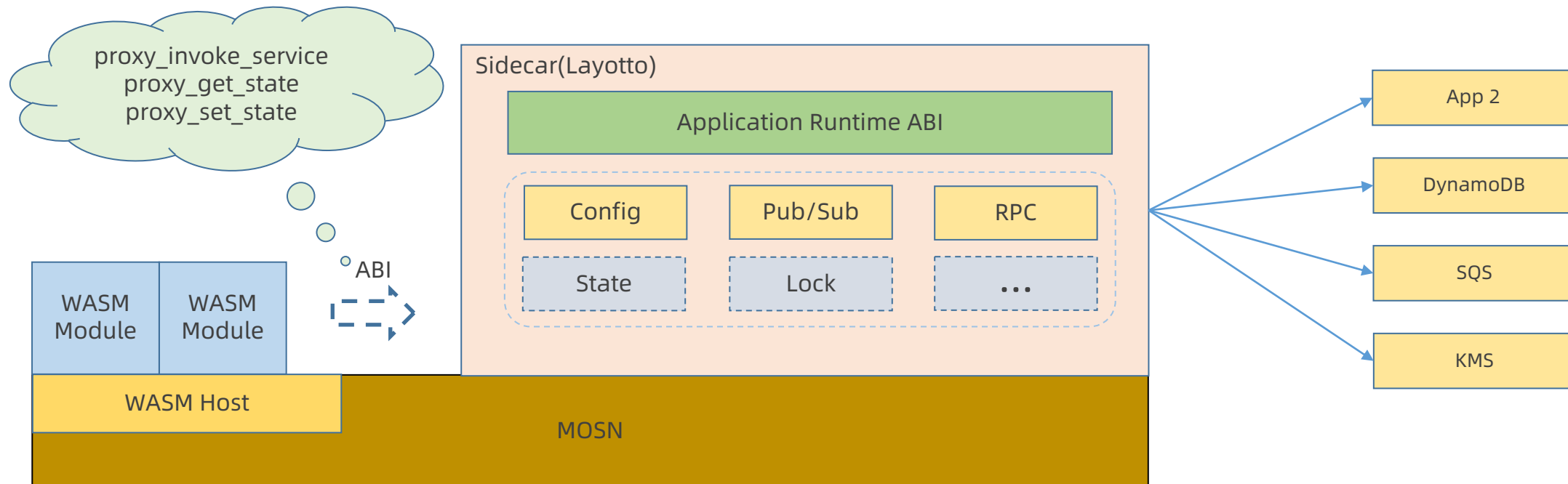
CLOUD NATIVE
Wasm DAY
NORTH AMERICA



FaaS Exploration



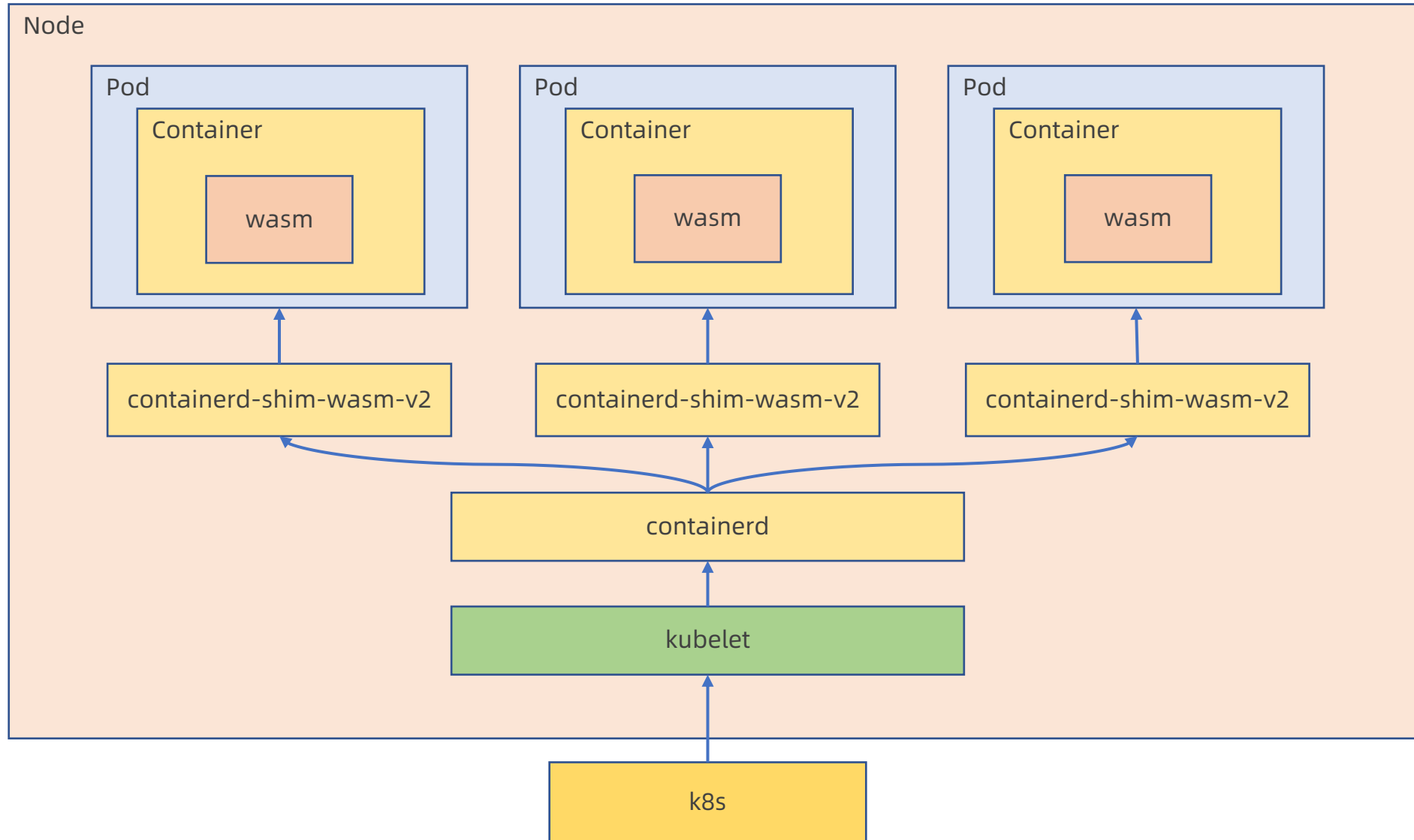
CLOUD NATIVE
Wasm DAY
NORTH AMERICA



Proof of Concept



CLOUD NATIVE
Wasm DAY
NORTH AMERICA

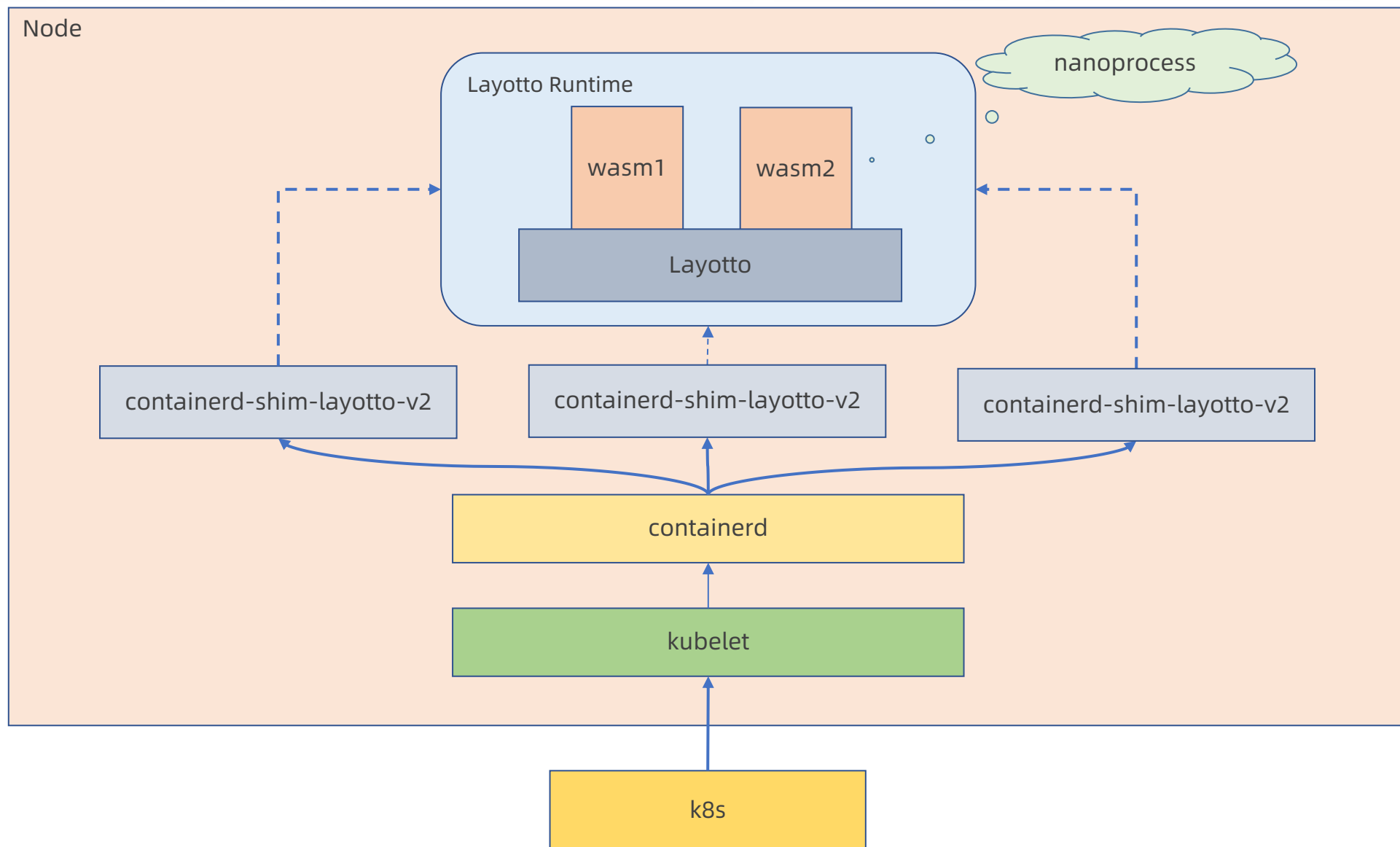


Reference: <https://github.com/dmccgowan/containerd-wasm>

Proof of Concept



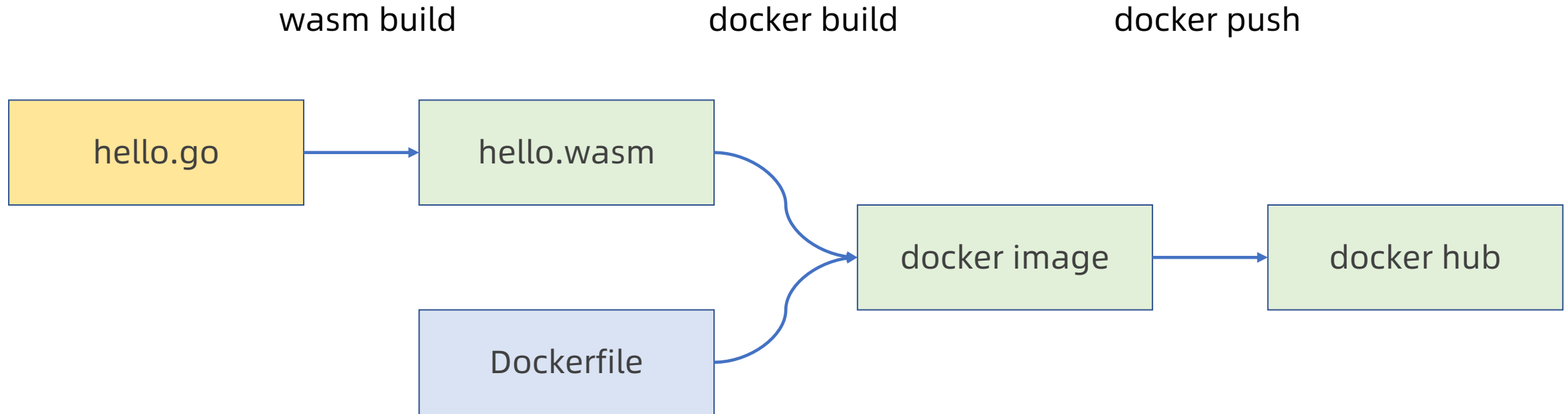
CLOUD NATIVE
Wasm DAY
NORTH AMERICA



Proof of Concept



CLOUD NATIVE
Wasm DAY
NORTH AMERICA



FROM scratch
COPY hello.wasm hello.wasm
CMD ["hello.wasm"]

Proof of Concept



CLOUD NATIVE
Wasm DAY
NORTH AMERICA

```
layotto > demo > wasm > code > golang > client > function1.go
function1.go x function2.go x
75 //
76 // function1 receive http request
77 // return book info
78 func (ctx *myHttpContext) OnHttpRequestBody(bodySize int, endOfStream bool) types.Action {
79     //1. get request body
80     body, err := proxywasm.GetHttpRequestBody(start: 0, bodySize)
81     if err != nil {
82         proxywasm.LogErrorf(format: "GetHttpRequestBody failed: %v", err)
83         return types.ActionPause
84     }
85
86     //2. parse request param
87     bookName, err := getQueryParam(string(body), paramName: "name")
88     if err != nil {
89         proxywasm.LogErrorf(format: "param not found: %v", err)
90         return types.ActionPause
91     }
92
93     //3. request function2 through ABI
94     inventories, err := proxywasm.InvokeService(id: "function2", method: "", bookName)
95     if err != nil {
96         proxywasm.LogErrorf(format: "invoke service failed: %v", err)
97         return types.ActionPause
98     }
99
100     //4. return result
101     proxywasm.SetHttpResponseBody([]byte("There are " + inventories + " inventories for " + bookName + "."))
102     return types.ActionContinue
103 }
104
105 func getQueryParam(body string, paramName string) (string, error) {
106     kvs := strings.Split(body, sep: "&")
107     for _, kv := range kvs {
108         param := strings.Split(kv, sep: "=")
109         if param[0] == paramName {
110             return param[1], nil
111         }
112     }
113     return "", errors.New("param not found")
114 }
115
116 *myHttpContext.OnHttpRequestBody(bodySize int, endOfStream bool) types.Action
```



WebAssembly + Application Runtime = A New Era of FaaS?



WebAssembly + Application Runtime = A New Era of FaaS!



CLOUD NATIVE
Wasm DAY
NORTH AMERICA

Thanks!

<https://github.com/mosn/layotto>