Integration as Code

Mifan Careem
VP - Solutions Architecture, WSO2
• Mifan Careem, Vice President of Solution Architecture @ WSO2
• Oversees Solution Architecture and customer facing technical interactions at WSO2
• Has worked with 200+ customers in API Strategy, Integration strategy and API led digital transformation initiatives
### WSO2 API-Led Integration Platform

Open Source API Management, Integration, Identity

**Start with API management...**

**API MANAGER**
- API design, creation, reuse, governance, and analytics
- 20K APIs for 200K orgs
- API analytics
- API designer
- API gateway
- API microgateway
- API publisher
- API storefront/marketplace
- API repository/registry
- API Analytics

**ENTERPRISE INTEGRATOR**
- Quick, iterative integration of any app, data, or system
- 6 trillion transactions / yr
- ESB
- Integration designer
- Message broker
- Workflows
- Business rules
- Streaming engine
- Stream processing
- Integration analytics

**IDENTITY SERVER**
- Secure and federated identity
- For APIs and integration
- 60M identities managed
- Identity management
- Identity federation / SSO
- Identity bridging
- API and microservices security
- Strong and adaptive Auth
- Access control
- Privacy control
- IAM and security analytics

Complement APIs with integration, security and analytics to connect apps and data.
From Integration Imperative to Exploding Endpoints
Disaggregated architectures drive 50 billion endpoints, growing >1 trillion

Scale and agility are pushing app disaggregation...

...that makes hybrid integration the unspoken challenge of all cloud services
Disaggregation leads to more endpoints

Everything is An Endpoint

- Functions
- APIs
- Data
- SaaS
- Legacy
- Devices

PROGRAMMABLE ENDPOINTS

DECADES

- 1970s
- 1980s
- 1990s
- 2000s
- 2010s
- 2020+
Integration in an increasingly disaggregated world

CLOUD NATIVE APPS

Transactions
Circuit Breaking
Protocols
Payloads
Events
Security
Workflow
Streams
Compensation
The Integration Gap

INTEGRATION PRODUCTS
ESB, BPMN, EAI

NOT AGILE

GENERAL PURPOSE PROGRAMMING LANGUAGES
Java / Spring
JavaScript / Node

NOT INTEGRATION SIMPLE
Through 2020, integration work will account for 50% of the time and cost of building a digital platform.

Use a Hybrid Integration Approach for Digital Transformation

50% of software development will be integration
Ballerina
Cloud Native Programming Language

```ballerina
@kubernetes:Deployment{
    image: "corp/microsocial",
}

@apiGateway{
    security: "OAuth",
    transactionPerSec: 15
}

service http> myService {
    @http:ResourceConfig {
        methods: ["POST"]
    }
    resource(caller, request) {
        endpoint twitter:client t {
            transaction {
                t -> tweet(...);
                caller -> respond( ... );
            }
        }
    }
}
```
Ballerina is a compiled, type safe, Turing-complete, concurrent programming language.
Ballerina is the glue between microservices

Transactions
Circuit Breaking
Protocols
Payloads
Events
Security
Workflow
Streams
Compensation
Ecosystem extensions for disaggregated architecture

- Kubernetes
- Docker
- Jaeger
- Prometheus
- etcd
- Istio / Envoy
- OAuth2 / OIDC
- LinkerD
- Consul
- RabbitMQ
- AMQP
- Consul
- Swagger/OpenAPI
- MySQL
- gRPC / Protobuf
- Postgres
- WebSub
- Cassandra
- Spring/Java
- Visual Studio Code
- Intellij IDEA
- Ballerina API Gateway
- BALLERINA SERVICE
- BALLERINA BRIDGE
- BALLERINA MESSAGE BROKER
- BALLERINA TRANSACTION COORDINATOR
- BALLERINA IDENTITY BROKER
- ENDPOINTS
- observability
- security
**Ballerina bridges the Integration Gap**

<table>
<thead>
<tr>
<th>Discover at ballerina.io</th>
<th>Agile</th>
<th>Integration Simple</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Edit / Build / Run</td>
<td>Services</td>
</tr>
<tr>
<td></td>
<td>Language server</td>
<td>Endpoints</td>
</tr>
<tr>
<td></td>
<td>IDE plugins</td>
<td>Resources</td>
</tr>
<tr>
<td></td>
<td>Projects</td>
<td>Connectors</td>
</tr>
<tr>
<td></td>
<td>Docker and K8S</td>
<td>-&gt;</td>
</tr>
<tr>
<td></td>
<td>Debugger</td>
<td>Observability</td>
</tr>
<tr>
<td></td>
<td>Testerina: unit tests</td>
<td>CI/CD</td>
</tr>
<tr>
<td></td>
<td>Doc generation</td>
<td>Table, vector, map</td>
</tr>
<tr>
<td></td>
<td>CLI extensions</td>
<td>Struct</td>
</tr>
<tr>
<td></td>
<td>Dev tracing</td>
<td>Lambda</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>Tasks, scheduling</td>
</tr>
<tr>
<td></td>
<td>Projects</td>
<td>Dependency mgmt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Module management</td>
<td>Services</td>
</tr>
<tr>
<td></td>
<td>Ballerina Central</td>
<td>Endpoints</td>
</tr>
<tr>
<td></td>
<td>Type safety</td>
<td>Resources</td>
</tr>
<tr>
<td></td>
<td>Union types</td>
<td>Connectors</td>
</tr>
<tr>
<td></td>
<td>Flow control</td>
<td>-&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workers, fork/join</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message broker</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Versioning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bridge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Swagger</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Databases</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transformations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JSON primitive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Annotations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Circuit breaker</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Async</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gRPC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protobuf</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XML type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Streams</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CSV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Session mgmt</td>
</tr>
</tbody>
</table>
Composites for the enterprise
How to get involved

Learn more  http://ballerina.io

Open source  http://github.com/ballerina-platform/

Get support  Stack Overflow #ballerina tag

Learn about the company  http://wso2.com/
From ESBs to Smart Endpoints and Dumb Pipes

Microservices architectures promotes the elimination of a central ESB in places of smart endpoints connected by dumb pipes.
ESB as the central integration bus

- Centralized ESB layer provides integration and network communications along with governance
Smart endpoints and dumb pipes

- Microservices code has to take care of network communications and governance of services