Java Card Integration with the Oracle Hyperledger Fabric based Blockchain

Cristian TOMA

Oracle
Java Card and Embedded Security
JPG - Java Platform Group

December, 2022

Red Bull RACING ORACLE CLOUD PARTNER

Java™ ORACLE

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Java Card Blockchain

#agenda

01 | **Crypto Blockchain Technology**

Terminology, Architecture, Transactions, Wallets, …

02 | **Demo**

Oracle Hyperledger Fabric

03 | **Q&A**

Conclusions

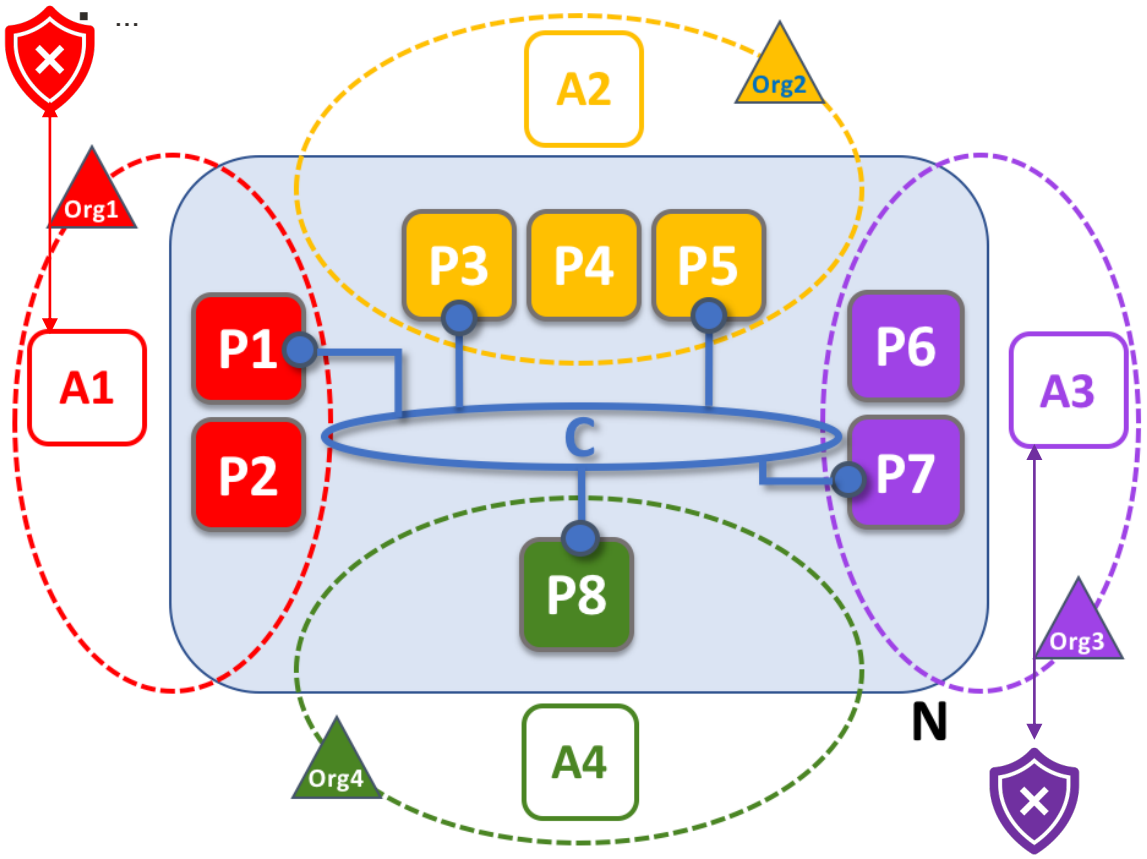
B-PaaS Clouds	Supported Blockchain Frameworks	Cloud Deployment and Interop. Maturity	Scalability and Security	E.U. Use Cases
Oracle	**Hyperledger Fabric**	▪ On-premises, public, hybrid, clouds ▪ Interoperability available (MiPasa, DAML)	▪ Scalability on VMs ▪ Certificates for Identity ▪ Digitaly signed messages ▪ Built-in encryption protection	▪ Dechatlon: Loyality points tracking in retail ▪ CargoSmart: Shipping and Logistics ▪ HealthSync: Healthcare info tracking ▪ Certified Origins: Food provenance
AWS	▪ **Hyperledger Fabric** ▪ Ethereum	▪ Running on AWS Cloud ▪ No Interoperability Info available	▪ Provides API for quick node creation ▪ AWS Key Management Service	▪ BMW Group: Auto Asset Provenance ▪ Nestle: Food origin Tracking
IBM	**Hyperledger Fabric**	▪ On-premises, public, hybrid, clouds ▪ Interoperability projects available	SecureKey Technologies	▪ iPoint Systems: Mineral provenance validation ▪ Nestle, Coop Italia: Food origin Tracking ▪ TradeLens: Trade and Shipping
Microsoft	▪ **Hyperledger Fabric** ▪ Quorum ▪ Ethereum ▪ Corda	▪ On-premises, public, hybrid, clouds ▪ Interoperability projects available	High with all Microsoft products: firewalls and TLS	▪ Crop Tracking ▪ Marine Insurance

Source IDC 2020: https://www.oracle.com/a/ocom/docs/corporate/analystrelations/idc-blockchain-europe.pdf

Java Card Secure Element Integration with Hyperledger Fabric and Terminology

Java Card Wallet Applets in Secure Element

- For transactions signature
- Interaction with Smart Contracts/Chain-code
- ...



N	Blockchain Network	L	Ledger
C	Channel	A	Application
P	Peer	PA, C	Principal PA (e.g. A1, P5) communicates via channel C.
		Org	Organization
A1, R, P1, P2			Organization R owns application A1 and peers P1, P2.

Java Card Wallet Applets in Secure Element

- For transactions signature
- Interaction with Smart Contracts/Chain-code
- ...

Copyright: https://hyperledger-fabric.readthedocs.io/en/latest/test_network.html | https://docs.oracle.com/en/cloud/paas/blockchain-cloud/usingoci/whats-console.html

Java Card Blockchain
#agenda

01 | **Crypto Blockchain Technology**

Terminology, Architecture, Transactions, Wallets, …

02 | **Demo**

Oracle Hyperledger Fabric

03 | **Q&A**

Conclusions

DEMO USE CASE

Blockchain Platform includes:

- A network of validating nodes (peers)
- Distributed ledger (linked blocks, world state and history DB)
- Ordering service (for creating blocks)
- Membership services (for managing organizations in a permissioned blockchain)

The Oracle Blockchain Cloud Chain-code Samples page contains:

• ***The Car Dealer sample*** includes a chain-code to manage the production, transfer, and querying of vehicle parts; the vehicles assembled from these parts; and transfer of the vehicles.

- In this sample, a large auto maker and its dealers and buyers have created a blockchain network to streamline its supply chain activities.
- Blockchain helps them reduce the time required to reconcile issues with the vehicle and parts audit trail.
- Blockchain client applications use Java Card applet for signing the blockchain transactions and invoking securely the smart contracts / chain-code from the Oracle Blockchain Platform-as-a-Service Cloud.

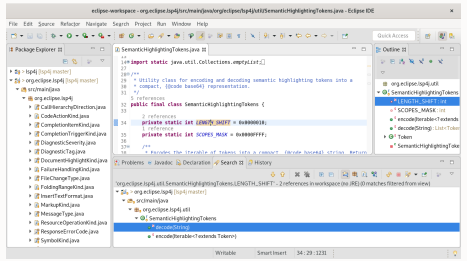
• ***The Fiat Money Token sample***
• ***The Balance Transfer sample***
• ***The Marbles sample***

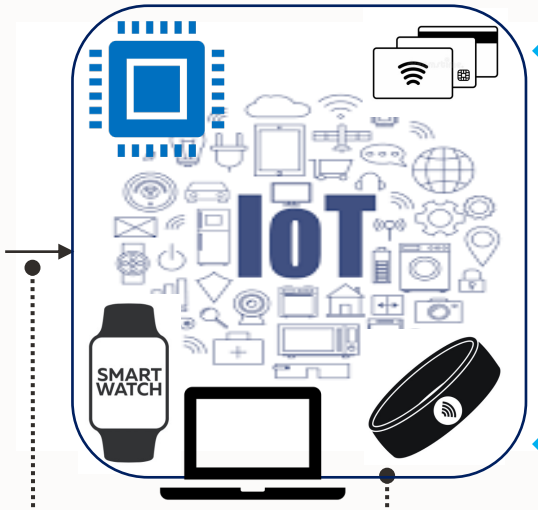
https://www.oracle.com/webfolder/s/assets/ebook/developing-dapps-oracle-blockchain/pdf/DevDApps.pdf | https://docs.oracle.com/en/cloud/paas/blockchain-cloud/usingoci/what-are-chaincode-samples.html

Java Card Blockchain Demo – High Level Components

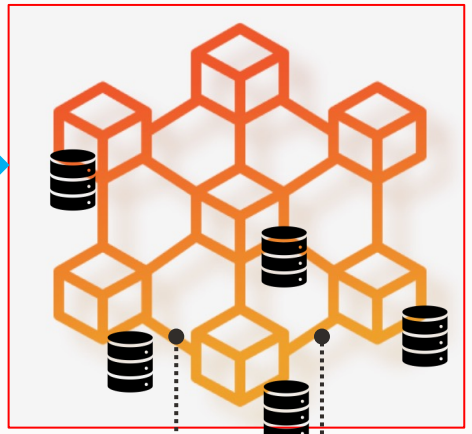
IoT/B-PaaS Cloud Auth:
ECDSA/RSA w SHA-256
IPSO, CBOR/COSE, JSON over
MQTTs or HTTPs

Java Card Dev Kit





HTTPs / WebSocket:
Blockchain RPC



Deploy Java Card
Applet to the device

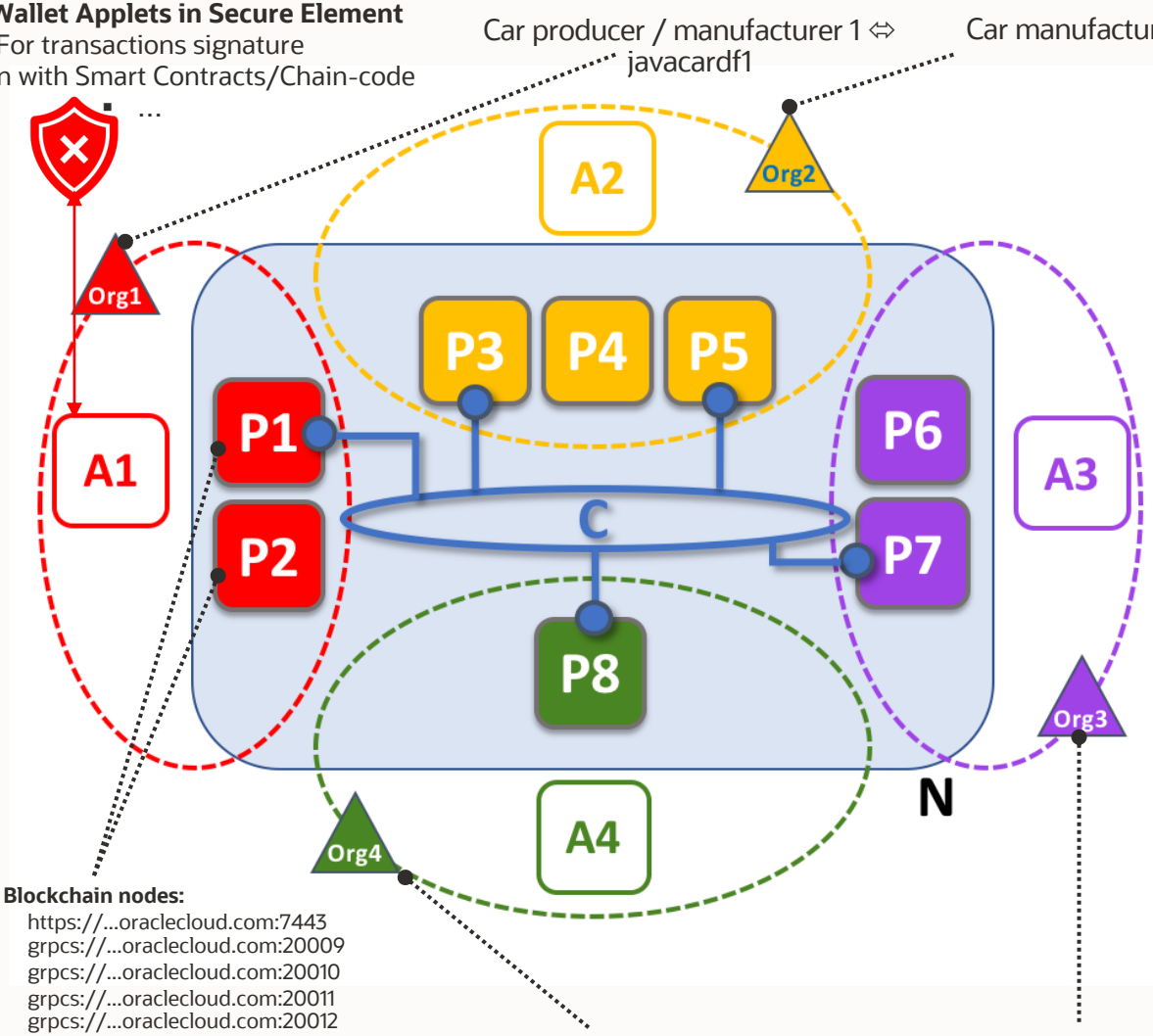
Secure Element with:
Java Card Runtime
Running JC Applets

Blockchain P2P
Protocols/Socket

Hyperledger Fabric v2 Terminology and Demo Mapping

DEMO: The Blockchain Car demo shows how a client application of the red orgranization (e.g. Car Manufacturer <=> javacardf1) is triggering a smart contract (chaincode) method via a channel shared with another organization (e.g. Car components suppliers).

Java Card Wallet Applets in Secure Element
- For transactions signature
- Interaction with Smart Contracts/Chain-code



N	Blockchain Network	L	Ledger
C	Channel	A	Application
P	Peer	PA, C	Principal PA (e.g. A1, P5) communicates via channel C.
		Org	Organization
A1, R, P1, P2			Organization R owns application A1 and peers P1, P2.

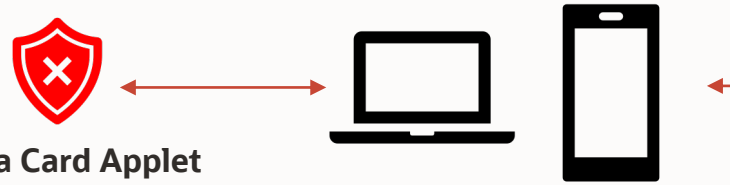
Blockchain nodes:

console	https://...oraclecloud.com:7443
peer0	grpcs://...oraclecloud.com:20009
peer1	grpcs://...oraclecloud.com:20010
peer2	grpcs://...oraclecloud.com:20011
peer3	grpcs://...oraclecloud.com:20012
orderer0	grpcs://...oraclecloud.com:20003
orderer1	grpcs://...oraclecloud.com:20005
orderer2	grpcs://...oraclecloud.com:20007
ca	https://...oraclecloud.com:7443
restproxy	https://.....com:7443/restproxy

Copyright: https://hyperledger-fabric.readthedocs.io/en/latest/test_network.html | https://docs.oracle.com/en/cloud/paas/blockchain-cloud/usingoci/whats-console.html

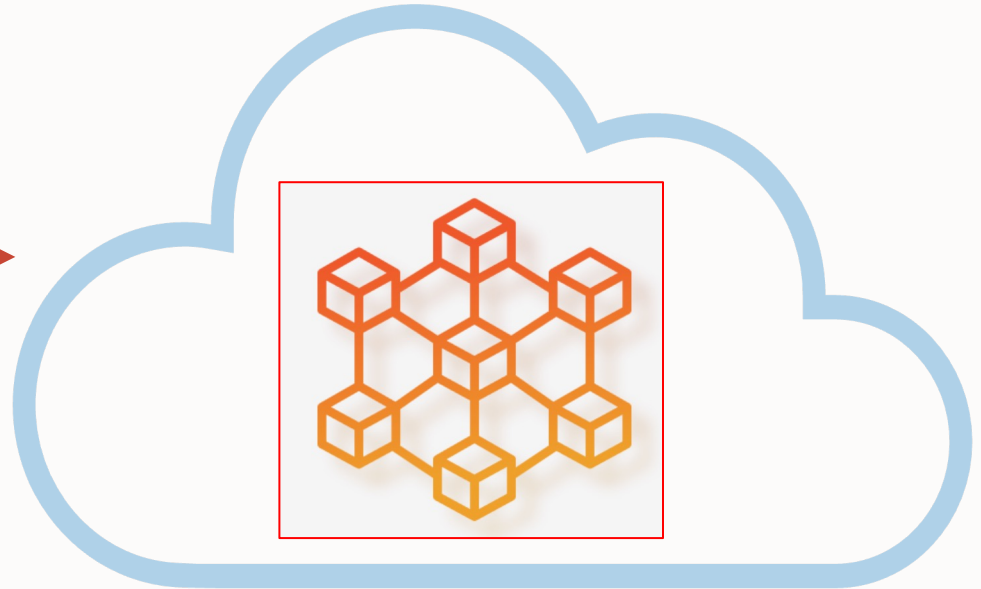
DEMO Setup - Java Card Integration with the Oracle Hyperledger Fabric based Blockchain

```
try (Gateway gateway = builder.connect()) {
  // Obtain a smart contract/chain-code deployed on the network.
  Network network = gateway.getNetwork(channelName);
  Contract contract = network.getContract(chaincodeName);
  // Create blockchain transaction and select smart contract business logic method.
  Transaction createCarTransaction = contract.createTransaction("initVehicle");
  // Submit blockchain transaction for interacting with the smart contract.
  byte[] createCarResult = createCarTransaction.submit("Oracle Red Bull F1", "Honda", ...);
```



Java Card Applet

Java Client Application
Hyperledger Fabric v2 Blockchain



Cloud: https://javacardf1-oabcs1-iad.blockchain.ocp.oraclecloud.com:7443/
Org: javacardf1

B-PaaS: Oracle Blockchain CS

```
// Java Card Applet
public class BlockchainCryptoWallet extends Applet {
...
short len = ECCUtils.sign (...);
apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA, (short) len);
...
```

Organization: javacardf1

ECDSA Certificate Public Key:

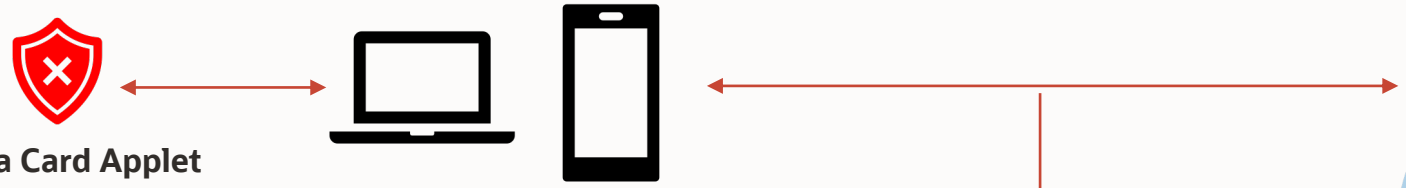
```
-----BEGIN CERTIFICATE-----
MIICKzCCAdG … Gw==
-----END CERTIFICATE-----
```

Blockchain binary message CONNECT

+ ECDSA SIGNATURE performed by Java Card Applet

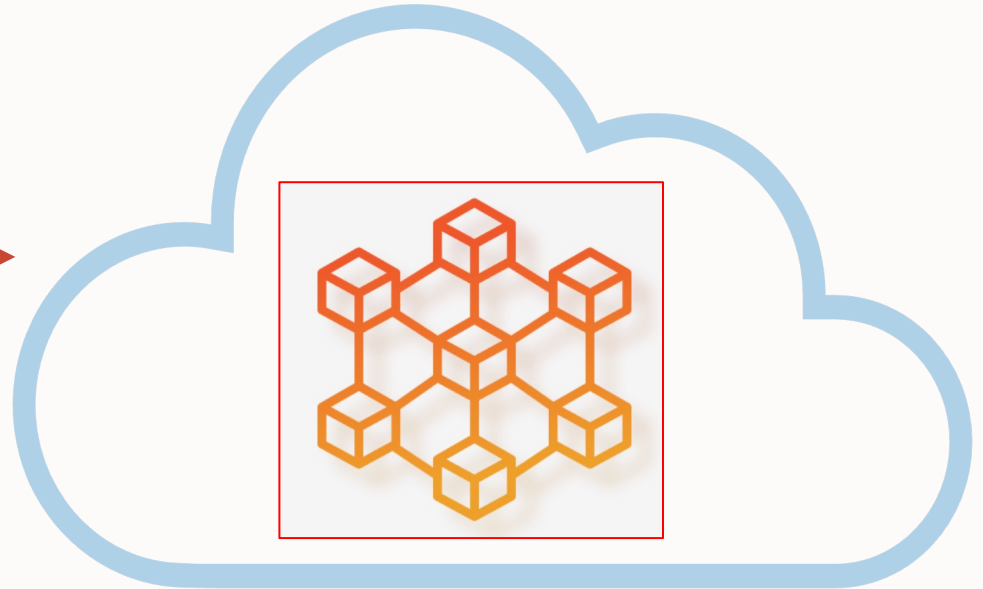
DEMO Setup - Java Card Integration with the Oracle Hyperledger Fabric based Blockchain

```
try (Gateway gateway = builder.connect()) {
  // Obtain a smart contract/chain-code deployed on the network.
  Network network = gateway.getNetwork(channelName);
  Contract contract = network.getContract(chaincodeName);
  // Create blockchain transaction and select smart contract business logic method.
  Transaction createCarTransaction = contract.createTransaction("initVehicle");
  // Submit blockchain transaction for interacting with the smart contract.
  byte[] createCarResult = createCarTransaction.submit("Oracle Red Bull F1", "Honda", ...);
```



Java Card Applet

Java Client Application
Hyperledger Fabric v2 Blockchain



Cloud: https://javacardf1-oabcs1-iad.blockchain.ocp.oraclecloud.com:7443/
Org: javacardf1

B-PaaS: Oracle Blockchain CS

```
// Java Card Applet
public class BlockchainCryptoWallet extends Applet {
...
short len = ECCUtils.sign (...);
apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA, (short) len);
...
```

Organization: javacardf1

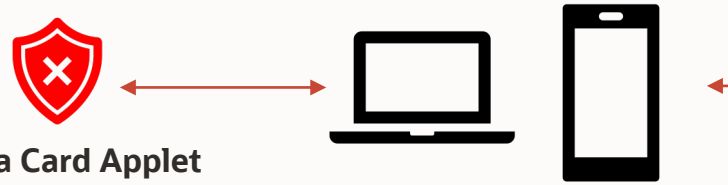
ECDSA Certificate Public Key:
```
-----BEGIN CERTIFICATE-----
MIICKzCCAdG … Gw==
-----END CERTIFICATE-----
```

Blockchain binary message
GET NETWORK CONNECTION
channelName="default"

+ ECDSA SIGNATURE performed by Java Card Applet

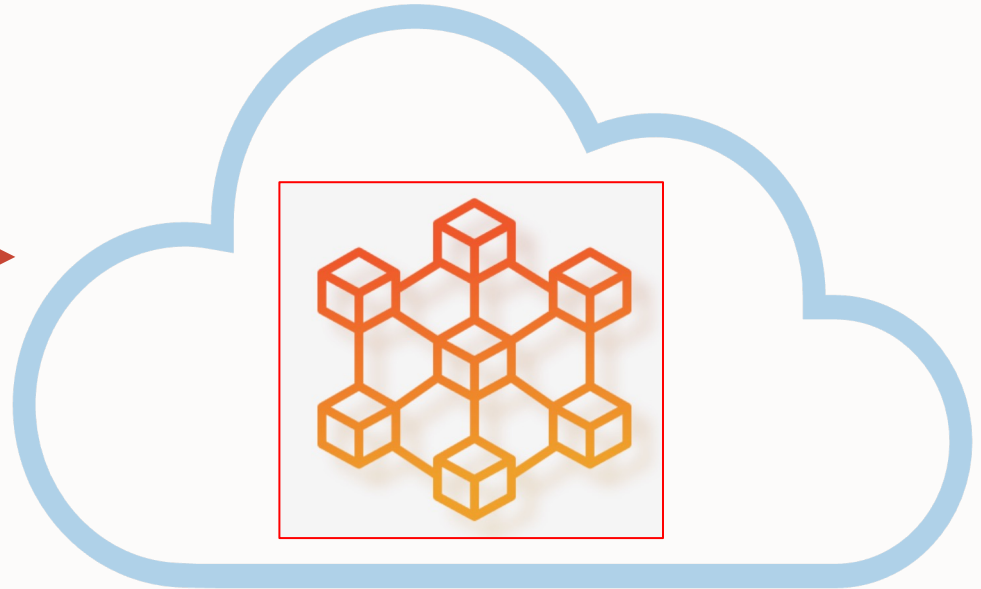
DEMO Setup - Java Card Integration with the Oracle Hyperledger Fabric based Blockchain

```
try (Gateway gateway = builder.connect()) {
  // Obtain a smart contract/chain-code deployed on the network.
  Network network = gateway.getNetwork(channelName);
  Contract contract = network.getContract(chaincodeName);
  // Create blockchain transaction and select smart contract business logic method.
  Transaction createCarTransaction = contract.createTransaction("initVehicle");
  // Submit blockchain transaction for interacting with the smart contract.
  byte[] createCarResult = createCarTransaction.submit("Oracle Red Bull F1", "Honda", ...);
```



Java Card Applet

Java Client Application
Hyperledger Fabric v2 Blockchain



Cloud: https://javacardf1-oabcs1-iad.blockchain.ocp.oraclecloud.com:7443/
Org: javacardf1

B-PaaS: Oracle Blockchain CS

```
// Java Card Applet
public class BlockchainCryptoWallet extends Applet {
...
short len = ECCUtils.sign (...);
apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA, (short) len);
...
```

Organization: javacardf1

ECDSA Certificate Public Key:

```
-----BEGIN CERTIFICATE-----
MIICKzCCAdG … Gw==
-----END CERTIFICATE-----
```

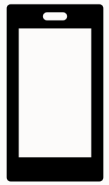
Blockchain binary message
OBTAIN SMARTCONTRACT by chaincodeName="obcs-cardealer-node:91..."

+ ECDSA SIGNATURE performed by Java Card Applet

DEMO Setup - Java Card Integration with the Oracle Hyperledger Fabric based Blockchain

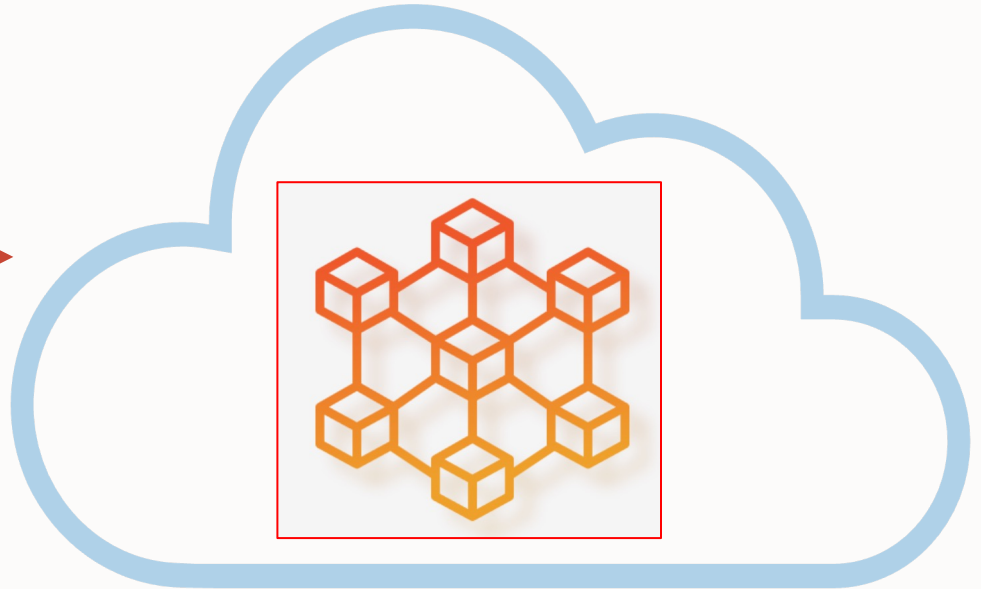
```
try (Gateway gateway = builder.connect()) {
  // Obtain a smart contract/chain-code deployed on the network.
  Network network = gateway.getNetwork(channelName);
  Contract contract = network.getContract(chaincodeName);
  // Create blockchain transaction and select smart contract business logic method.
  Transaction createCarTransaction = contract.createTransaction("initVehicle");
  // Submit blockchain transaction for interacting with the smart contract.
  byte[] createCarResult = createCarTransaction.submit("Oracle Red Bull F1", "Honda", ...);
```

Java Card Applet



Java Client Application
Hyperledger Fabric v2 Blockchain

Interaction with Blockchain via Signed Transactions for invoking Smartcontract/Chaincode business logic



Cloud: https://javacardf1-oabcs1-iad.blockchain.ocp.oraclecloud.com:7443/
Org: javacardf1

B-PaaS: Oracle Blockchain CS

```
// Java Card Applet
public class BlockchainCryptoWallet extends Applet {
...
short len = ECCUtils.sign (...);
apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA, (short) len);
...
```

Organization: javacardf1

ECDSA Certificate Public Key:

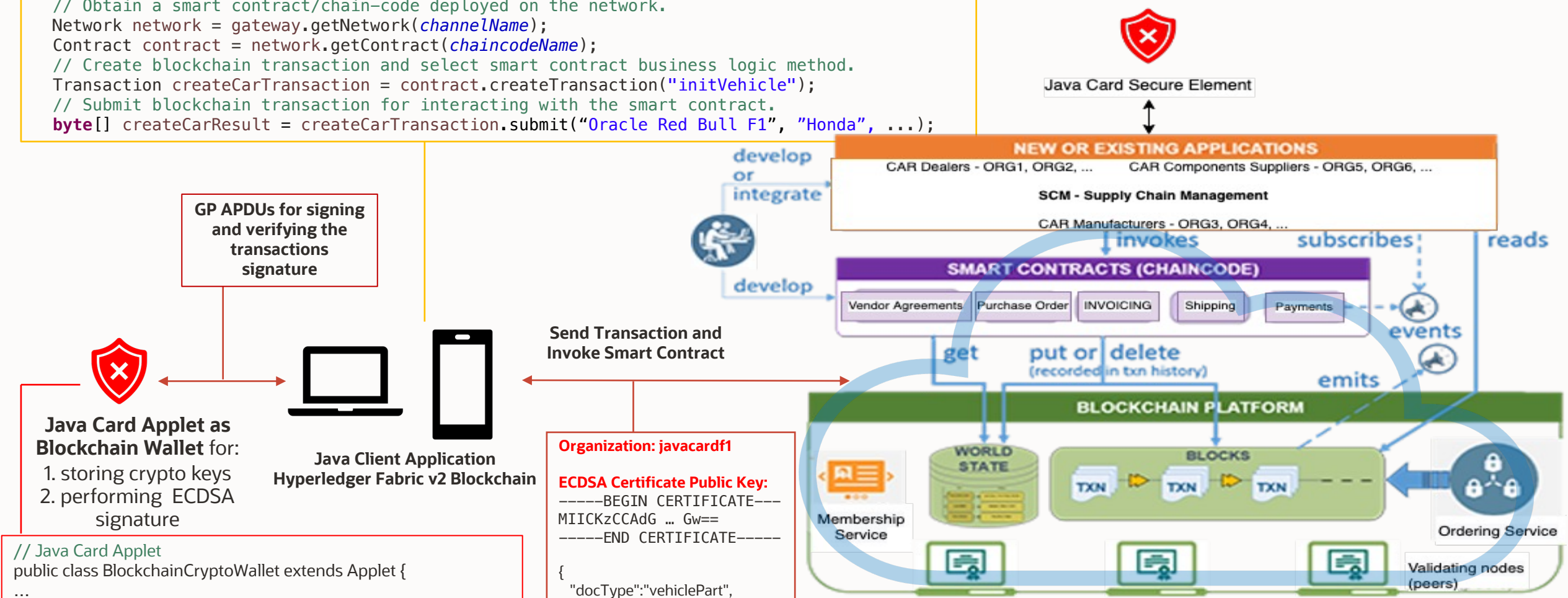
```
-----BEGIN CERTIFICATE---
MIICKzCCAdG … Gw==
-----END CERTIFICATE-----
```

```
{
  "docType":"vehiclePart",
  "serialNumber":"ser110002",
  "assembler":"tasa",
  ...
}
```

+ ECDSA SIGNATURE performed by Java Card Applet

DEMO Setup - Java Card Integration with the Oracle Hyperledger Fabric based Blockchain

```
try (Gateway gateway = builder.connect()) {
  // Obtain a smart contract/chain-code deployed on the network.
  Network network = gateway.getNetwork(channelName);
  Contract contract = network.getContract(chaincodeName);
  // Create blockchain transaction and select smart contract business logic method.
  Transaction createCarTransaction = contract.createTransaction("initVehicle");
  // Submit blockchain transaction for interacting with the smart contract.
  byte[] createCarResult = createCarTransaction.submit("Oracle Red Bull F1", "Honda", ...);
```



GP APDUs for signing and verifying the transactions signature

Java Card Applet as Blockchain Wallet for:
1. storing crypto keys
2. performing ECDSA signature

Java Client Application
Hyperledger Fabric v2 Blockchain

Send Transaction and Invoke Smart Contract

```
// Java Card Applet
public class BlockchainCryptoWallet extends Applet {
...
short len = ECCUtils.sign (...);
apdu.setOutgoingAndSend(ISO7816.OFFSET_CDATA, (short) len);
...
```

Organization: javacardf1

ECDSA Certificate Public Key:

```
-----BEGIN CERTIFICATE---
MIICKzCCAdG … Gw==
-----END CERTIFICATE-----
```

```
{
  "docType":"vehiclePart",
  "serialNumber":"ser110002",
  "assembler":"tasa",
   ...
}
```

+ ECDSA SIGNATURE performed by Java Card Applet

B-PaaS: Oracle Blockchain CS

Java Card Blockchain
#agenda

01 | **Crypto Blockchain Technology**

Architecture, Transactions, Wallets, …

02 | **Demo**

Hyperledger Fabric Oracle

03 | **Q&A**

Conclusions

Secure Blockchain Wallet using Java Card Technology and Platform

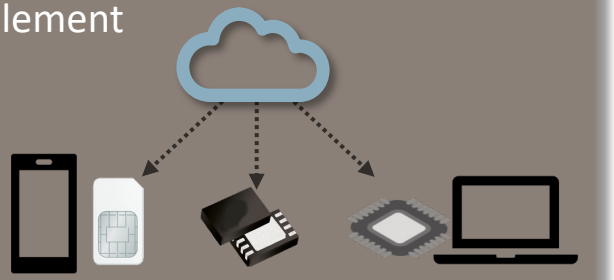
Conclusion

Secure Runtime

- Securely store and manage crypto keys for IoT Cloud and Blockchain Cloud Service Authentication
- Run the cryptographic algorithms in the Secure Element: create tokens, encrypt and sign the payload, blockchain transactions, …

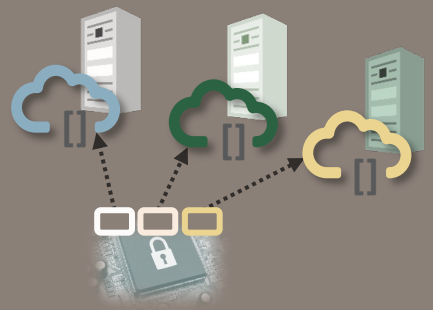
Portable

- Blockchain Secure Wallet is running on any Java Card enabled Secure Element
- Java Card Runtime agnostic to the hardware form factor of the secure element
- Running on any device (e.g. smart phone, laptop, authenticator dedicated device) hosting the blockchain client and the secure element



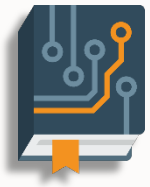
Adaptable & Extensible

- Support multiple application specific to the blockchain implementations e.g. Hyper-ledger Fabric and Ethereum wallets, Oracle Authenticator, IoT-Safe, FIDO, …
- Update and upgrade the Java Card applets to adapt to the fast evolving blockchain interface security requirements



More Information

https://www.oracle.com/java/technologies/java-card-tech.html

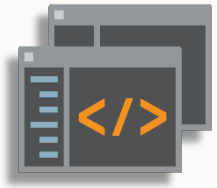


Java Card Platform Specification 3.1

Latest release of the Java Card specification and the reference for Java Card products.

Java Card Development Kit Tools

The Java Card Development Kit Tools are used to convert and verify Java Card applications. The Tools can be used with products based on version 3.1, 3.0.5 and 3.0.4 of the Java Card Specifications.



Java Card Development Kit Simulator

The Java Card Development Kit Simulator includes a simulation component and Eclipse plug-in.
Combined with the Java Card Development Kit Tools, it provides a complete, stand-alone development environment.

Java Card IoT and Security blog

This Blog covers the latest Java technology for small devices and security in the IoT, Mobile, ID and Payment.

Webcast – Secure Business Runs Java Card

Webcast – How to secure IoT Edge with Java Card

Webcast: Oracle Java Card 3.1 Boosts Security for IoT Devices at the Edge

contact: cristian.v.toma [at] oracle [dot] com | https://www.oracle.com/java/contact-form.html

ORACLE