Develop & Deploy
Hyperledger Fabric Apps
Using KrypCore

Hyperledger Fabric Layer 2 Platform
Key Features

Quick & Easy

Multiple Deployment Options

Cost Management & Optimization

Operation / Governance / Security

Maintenance
Quick & Easy Hyperledger Fabric Starter

Interactive GUI driven feature for

- Network Creation
- Network interaction
- Chaincode & Channel Management
- Operation & Administration
- Identity Management - Wallet

No Prerequisites:

- Docker, Docker Compose YAML
- Fabric Tools, Fabric Commands
- Linux, Shell

Expected Skill Set

- Chaincode Coding
- Rest API – Consume using Preferred Programming Language
- Knowledge of Endorsement Policies
Simplified & Focused

Focus on use case design, development and logic validation

Abstracts Hyperledger Fabric administration, configuration

Knowledge of Rest API is enough to perform transaction

Fabric SDK’s experience not required

Simplified Report Generation

Notification Service provides easy alerts and messages integration
Deploy On

Virtual Machines
- Docker Supported
- Linux, Windows
- For development
- For production

Managed Kubernetes Clusters
- Azure AKS
- AWS EKS
- Private Clouds
Deploy For

Development
- Design use case
- Perform Integrations
- Deploy Chain Codes

POC / Pilot
- Validate

Production
- Manage & Scale
Deploy Via

- GUI – Super Easy
  - Simple & Interactive. Configure cloud resources & execute
  - Ideal for anyone – Cloud Skills not required

- Terraform, Helm Charts, Scripts – Power Users
  - Powerful, seamless deployment via commands and automation frameworks
  - Ideal for enterprises, devops, secOps engineer

- ARM Templates – Azure Ready
  - Simple & Interactive. Seamless Azure native deployment
  - Ideal for Azure ARM
Cost Optimized Blockchain

• Project Execution shortened by $\frac{1}{3}$rd time & $\frac{1}{3}$rd Cost
• Forecast Infrastructure Upgrade – Using data lake size analysis
• Reduced Integrations: Leverage on API & Email Notifications – Pub Sub
• Reduced report development time frame: Using data lake report builder
• Reduced Administration Charges: Fabric Administration simplified - cutting down the deep learning curve of Hyperledger Fabric and its nuances
• Low Code Option – No chaincode development Required
• Optimized cloud resource usage with tried and tested configurations provides lower consumption fees.
Stratified user management where each strata has pre-defined privileges within the network and the application.

User blocks with limited permissions in blockchain operations.

Segregation of deployments at Kubernetes level via namespaces.

Certificates and Keys managed using tried and tested cloud solutions for key management.
Maintenance

- Easy maintenance of production systems with top-of-the-line open-source monitoring tools leveraging Hyperledger Metrics.
  - Prometheus – Grafana
- Alerts set for blockchain data and activity.
- Alert set for Kubernetes dashboard for cluster resource monitoring.
  - Kubernetes Dashboard
- Alerts set of cloud infrastructure monitoring for costing and control of resources.
- Log Monitoring and Analytics.
  - Azure Log Analytics Workbook
  - Elasticsearch-FluentD-Kibana
End of Presentation

Thank You
Support Slides
KrypCore vs. Bespoke

Development Comparison

**BLOCKCHAIN SETUP**
- Set up, Docker
- Docker Compose
- Golang | Node JS, IDE
- Download and install Hyperledger using docker image file
- Fabric CA setup

**CONFIGURATION**
- Configure Fabric SDK to interact with your application
- Selection and configuration of IDE based on your business requirement

**APPLICATION CONFIGURATION**
- Configure PEER, ORDERER, CA, ORGANIZATION and CHANNELS

**CHAINCODE MANAGEMENT**
- Install, Instantiate - Business logic coding
- Visualization of chaincode activity, post analysis of transaction of blockchain

**USER INTERFACE**
- Custom user interface to manage and interact the application

**KRYPCORE SANDBOX**

- BLOCKCHAIN SETUP
- CONFIGURATION
- APPLICATION CONFIGURATION
- CHAINCODE MANAGEMENT
- GRAPHICAL USER INTERFACE
- MULTIPLE USECASES
**KRYPCORE DATA LAKE**

Enables high performance searches of the Blockchain data. Component facilitates to configure business data that needs to be indexed. Block data will be synchronized periodically with index database. As part of this process business data will be extracted from the transaction payload and maintained in database to support BI tool.

- BUSINESS INTELLIGENCE TOOL
- CUSTOM APPLICATION
- HIGH PERFORMANCE OF BLOCKCHAIN DATA
- BUSINESS DATA VISUALIZATION
- OPEN DATA API FOR INNOVATION
Notification Services

**EVENT PUBLISH / SUBSCRIBE**

Notification for participants upon occurrence of an event/condition in the Blockchain network allows controlling with ease, the event sequences enabling better notification management for users.
Chaincode Life-Cycle Management
Manage chaincode lifecycle methods
Chaincode Lifecycle Management

- Chaincode Packaging
  - Supports GoLang / JS / Java Chaincode Packaging
  - Package supported for 1.4 and 2.2 formats
- Installing on Peers
- Approve, Committing
  - Perform approvals and commit of new versions
- View Configuration
  - View existing configuration of the chaincodes
- Define PDC Config
  - Defined PDC Collections (Explicit)
- Define Endorsement Policies
  - Define Custom Endorsement Policies
Transaction & Query API’s

- REST based HTTP API’s for Invoke & Query
- JSON Payload
- Payload Attributes
  - channel name,
  - chaincode name,
  - function name and
  - CC Arguments.
  - Authentication Params
- All arguments are strings
- Optional Attributes:
  - Proxy
  - Endorsers List
- Automatic Endorser Selection is enabled if no endorsers are provided
- API Authentication based on:
  - Wallet Credentials Or
  - Authentication token
EXECUTE CHAINCODE

Invoke chaincode events

POST /api/network.invoke

Parameters

```
{  "function": "initLedger",  "arguments": [    {      "string": "default"    }  ],  "transient": {    "key1": "value",    "key2": 10  }}
```

BODY SAMPLE

```
{
  "chaincodeId": "tworgcc",
  "function": "initLedger",
  "arguments": [
    {      "string": "default"    }
  ],
  "transient": [
    {      "object": {}    }
  ]
}
```

RESPONSE

```
{
  "Status": true,
  "Msg": "success",
  "Extra": "fffa3e60eb5652a60e12ffed1af3656b2f02a879a09cde0e59af608f6867d39"
}
```

Response Messages

200 Chaincode successfully invoked.

default unexpected error.
Network Governance

Managing Channel Attributes
Channel Manager

• Allows Organization to manage the channel parameters in production
• Work-Flow driven UI to assist multiple members to sign and agree on the change
• Onward members can see / analyze the changes in the request file
• Simple changes like batch size to addition of a new organization can be performed
• Can be customized to support different file formats of other providers
Channel Manager

- Modify Anchor Peer List
- Modify Organization Admins
- Update CRL
- Add new channel
- Join new channel
- Change batch size, block time
- Change Policies
Channel Manager –
Upgrading Fabric Version

• Hyperledger version upgrades are activated using the capabilities attributes.
• This must be performed by majority of the members.
• Once done, the features of newer version are activated in the protocol
Identity Management

Create Identities & Manage lifecycle
Registrar & Workbench Administration

- Each Organization have registrar interface for managing organization user identities
  - Identities are created for
    - Performing transactions in blockchain
    - Managing the workbench
    - Network Governance
  - Multiple Affiliations can be created and managed
  - Revocation & CRL Generation is also available
User Wallet

• Each user / application can access the wallet to
• Manage credentials & access tokens
• View transactions
• Download private key and fabric folders for SDK integrations (if enabled)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST API’s</td>
<td>Easy to integrate JSON Payload</td>
</tr>
<tr>
<td>Notifications (Pub-Sub)</td>
<td>Email, REST End Points</td>
</tr>
<tr>
<td>Authentication</td>
<td>Wallet Credentials, Token</td>
</tr>
<tr>
<td>Data Explorer &amp; Reporting</td>
<td>Visualize data, Full state, world state and PDC</td>
</tr>
<tr>
<td>Manage Channel</td>
<td>Anchor Peer, Block Properties</td>
</tr>
</tbody>
</table>