

# Introduction to Fabric Composer

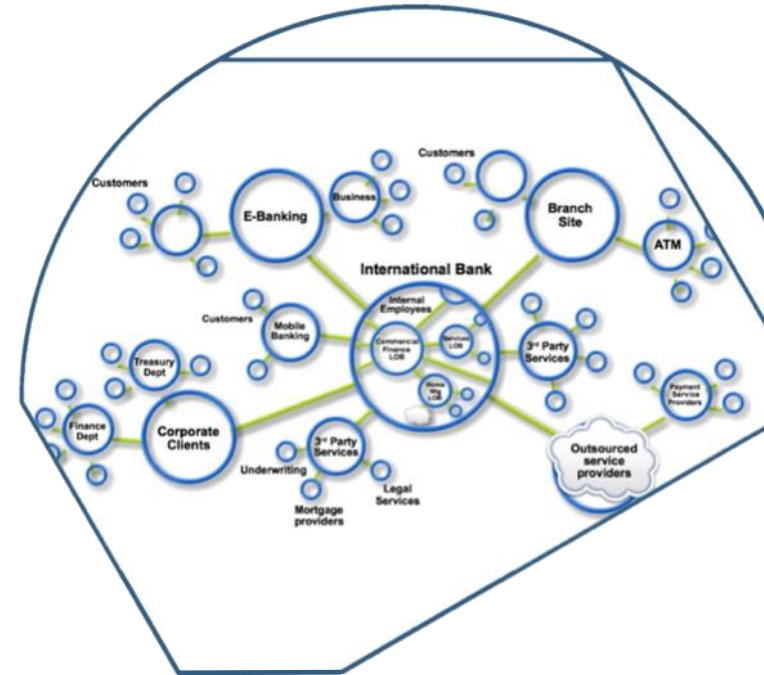
## WebSphere User Group

Matthew White  
Graham Charters

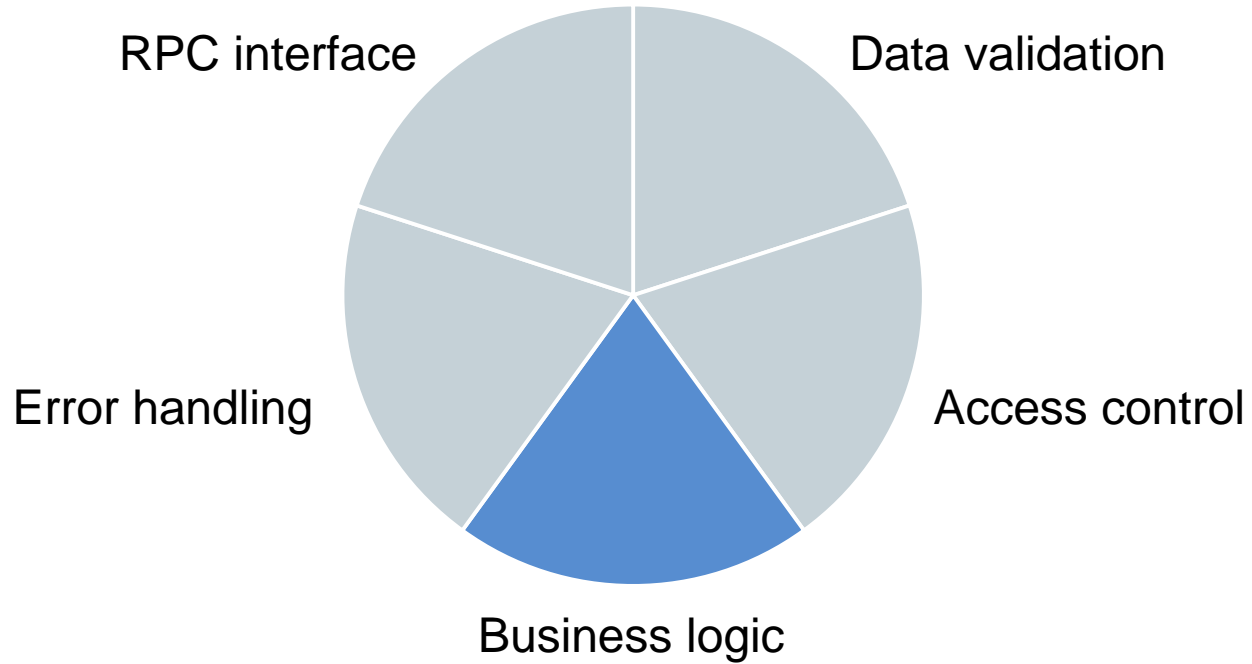


# Blockchain for business

- Blockchain builds on basic business concepts
  - **Business Networks** connect businesses
  - **Assets** flow over business networks
  - **Transactions** describe asset exchange
  - **Participants** submit transactions
  - **Contracts** define the rules for transactions
  - The **ledger** is a log of transactions
- Blockchain provides a shared, replicated ledger
  - Consensus, immutability, finality, provenance



# Hyperledger Fabric



# Hyperledger Fabric & Fabric Composer

- **Hyperledger Fabric** provides a technical foundation for transactional applications across business networks:
  - **Smart contracts** (chaincode) are executed on a distributed network
  - **Inputs** go into an immutable ledger; **outputs** go to a data store (world state)
  - Applications are built from the ground up
- **Fabric Composer** is a framework to accelerate the development of applications Start from the business level; model network **assets, participants, and transactions**
  - Applications use business centric APIs to invoke transactions that create, delete, and update assets and transfer them between participants
  - Assets, participants, and transactions are recorded in the world state in **registries**
  - Easily **integrate** Fabric with existing business processes and systems of record
  - Emphasis on quick solution creation and business-centric vocabulary

# Key development concepts

- **Model files** describe the **assets**, **participants**, and **transactions** in a business network
  - Expressive modelling language includes relationships, arrays, and validation rules
  - Data serialized as JSON, and is fully validated by Fabric Composer runtime
- **Access control lists** define rules for sharing and privacy
  - Rules automatically enforced by Fabric Composer runtime
- **Transaction processors** implement additional business requirements
  - Standard JavaScript code executed on the Fabric network by Fabric Composer runtime
- A **business network definition** is the set of the above for a given business network.



## Fabric Composer



## Hyperledger Fabric

```
.cto
Asset:
  hat
Participants:
  haberdasher, customer
Transaction:
  BuyAHat
```

A **model file** defines the **structure** of your business network in terms of Assets, Participants and Transactions

```
.js
function
onTransaction( ) {
  asset.owner =
  asset.new-owner;
}
```

A **script file** specifies how your Transactions will invoke, using Javascript

```
.acl
rule Default {
  participant: "ANY"
  operation: ALL
  action: DENY
}
```

An **Access Control List (ACL)** defines the rules that govern participants' ability to perform actions



Fabric Composer lets you package up your Business Network Definition into a **Business Network Archive** file  
**Pro Tip:** we like to pronounce it "banana file" ;)

Hyperledger Fabric v0.6

Hyperledger Fabric v1.0\*

Web Browser

Add **Connection Profiles** to let Fabric Composer deploy your Business Network Definition to a distributed ledger. For production use, we currently support **Hyperledger Fabric v0.6** (and v1.0 support is \*coming soon). We also offer a Web Browser profile for testing in your browser.

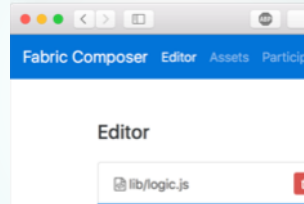
# Extensive, familiar, and open development toolset

```
asset Animal identi
  o String animal
  o AnimalType sp
  o MovementStatu
  o ProductionTyp
```

Data modelling



JavaScript  
business logic



Web playground

```
composer-client
composer-admin
```



Client libraries



Editor support

```
$ composer
```

CLI utilities



Code generation

Powered by  
  
LoopBack  
Node.js Framework



Integration

# Open means open

- Everything we have built so far has been open sourced! (Apache-2)
- All our development plans can be tracked and viewed in public GitHub
- We will be starting to run meetings in public for anyone to join
- We would love for you to come along and participate!
  - Review and comment on our future user stories
  - Contribute code by submitting pull requests
- Finally, we ~~will be submitting a proposal to~~ **have had a proposal accepted** by the Hyperledger Project TSC

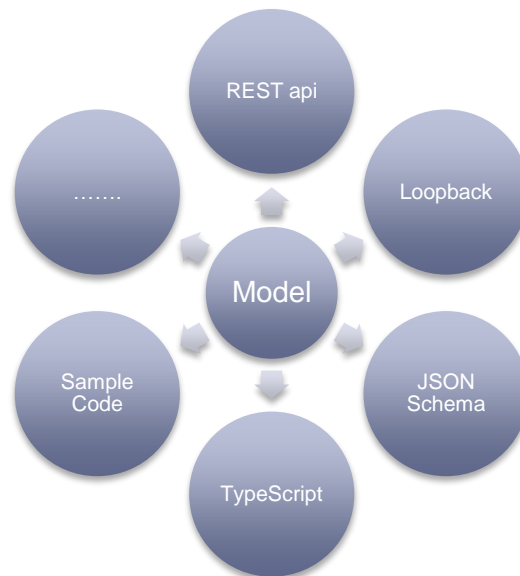


# Nothing can exist in isolation

- As business can't operate in isolation neither can software
- How does Fabric Composer help to integrated with <my software>??

- **APIs and Models**

- Using the definition of the model we can generator for example a Business Specific REST API
- Seed applications can be generated e.g. CLI, Angular-2

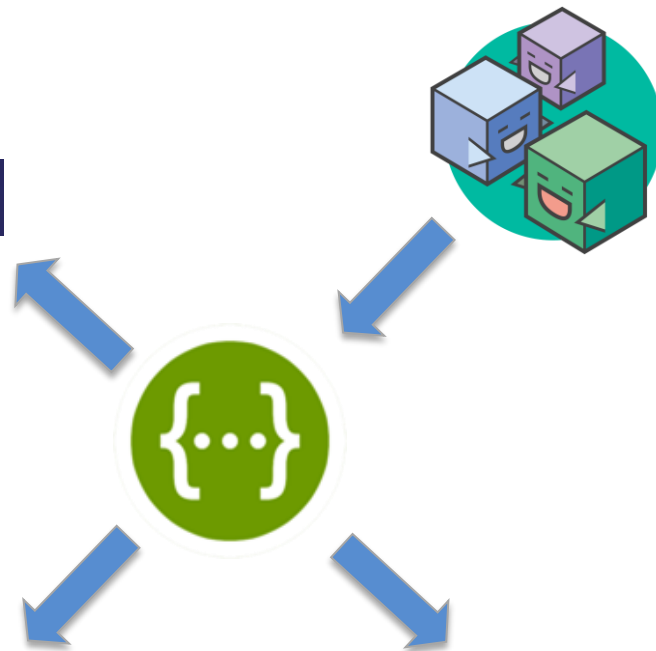




# Demo

- {{demo}}

IBM Integration Bus



# Links

- We're all over the internet!
  - Website, tutorial, and docs: <https://fabric-composer.github.io/>
  - Public demo: <http://fabric-composer.mybluemix.net/#/editor>
  - Source code: <https://github.com/fabric-composer/fabric-composer/>
  - NPM modules: <https://www.npmjs.com/search?q=fabric-composer>
  - JSDoc: <https://fabric-composer.github.io/jsdoc/develop/index.html>

# Community

- You can come and speak to us as well!
  - Chat to us using Rocket.Chat at <https://chat.hyperledger.org>
    - #fabric-composer and #fabric-composer-dev channels
  - Ask (and answer!) questions on Stack Overflow using the [fabric-composer](#) tag
  - Tweet us using the [#FabricComposer](#) hashtag!

Thank you!

