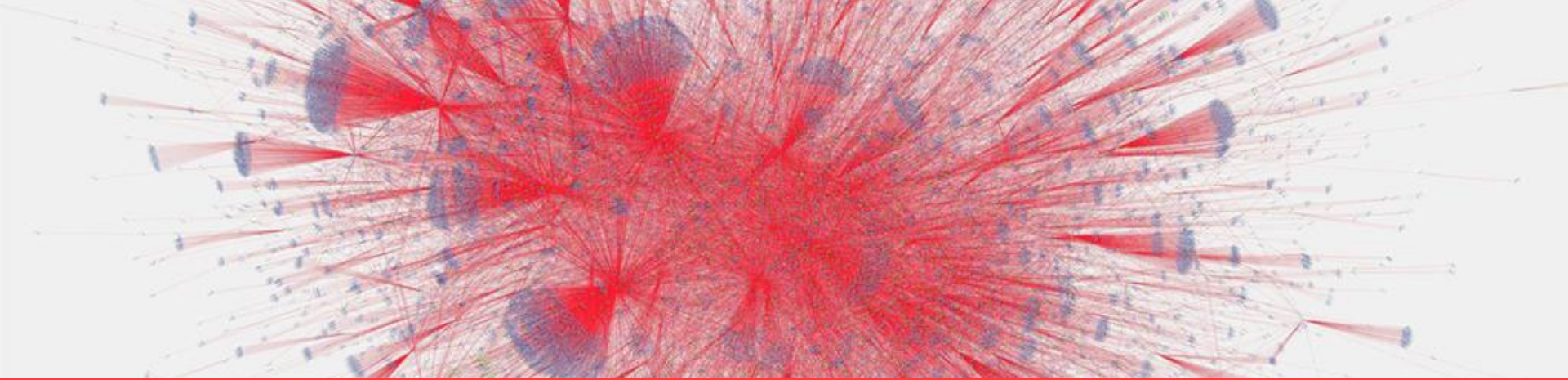




IBS IT COMPASS 2016

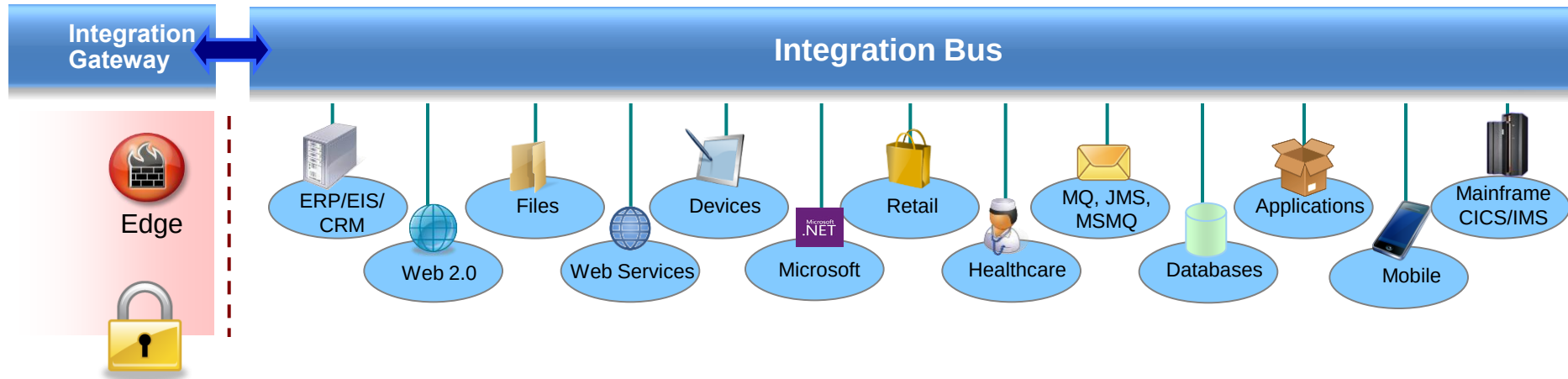
IBM INTEGRATION BUS

**Technical Overview And
Microservices**

Valentin Hristov
v.hristov@ibs.bg

Какво е IBM Integration Bus

- IBM Integration Bus представлява шина за услуги (ESB), която предоставя свързаност и унифициран модел за трансформация на данни в хетерогенна IT среда.



- IBM Integration Bus е новото име на WebSphere Message Broker

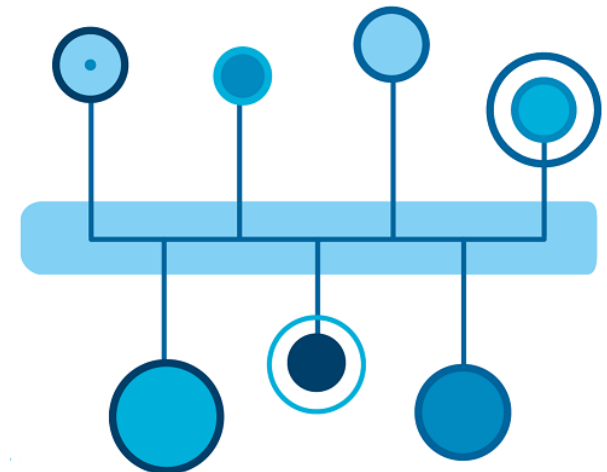
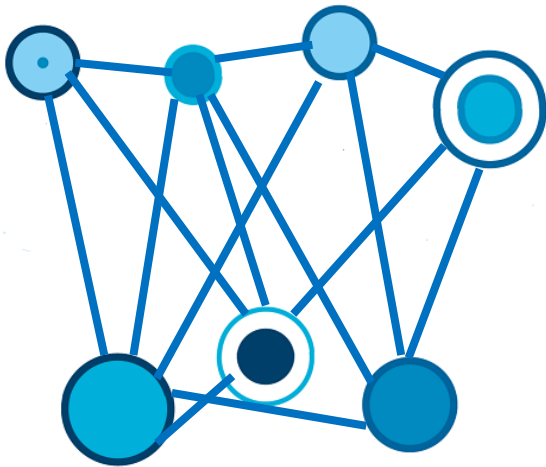
- Технология която е развивана над 15 години, използвана от над 2500 клиента по целият свят
- Посредством IIB бизнеса може да елиминира директните връзки между отделните системи и приложения, **без значение от платформата, протокола и формата на данните**.
- Всички интеграци биват централизирани, което многократно улеснява поддръжката.

Как помага IBM Integration Bus

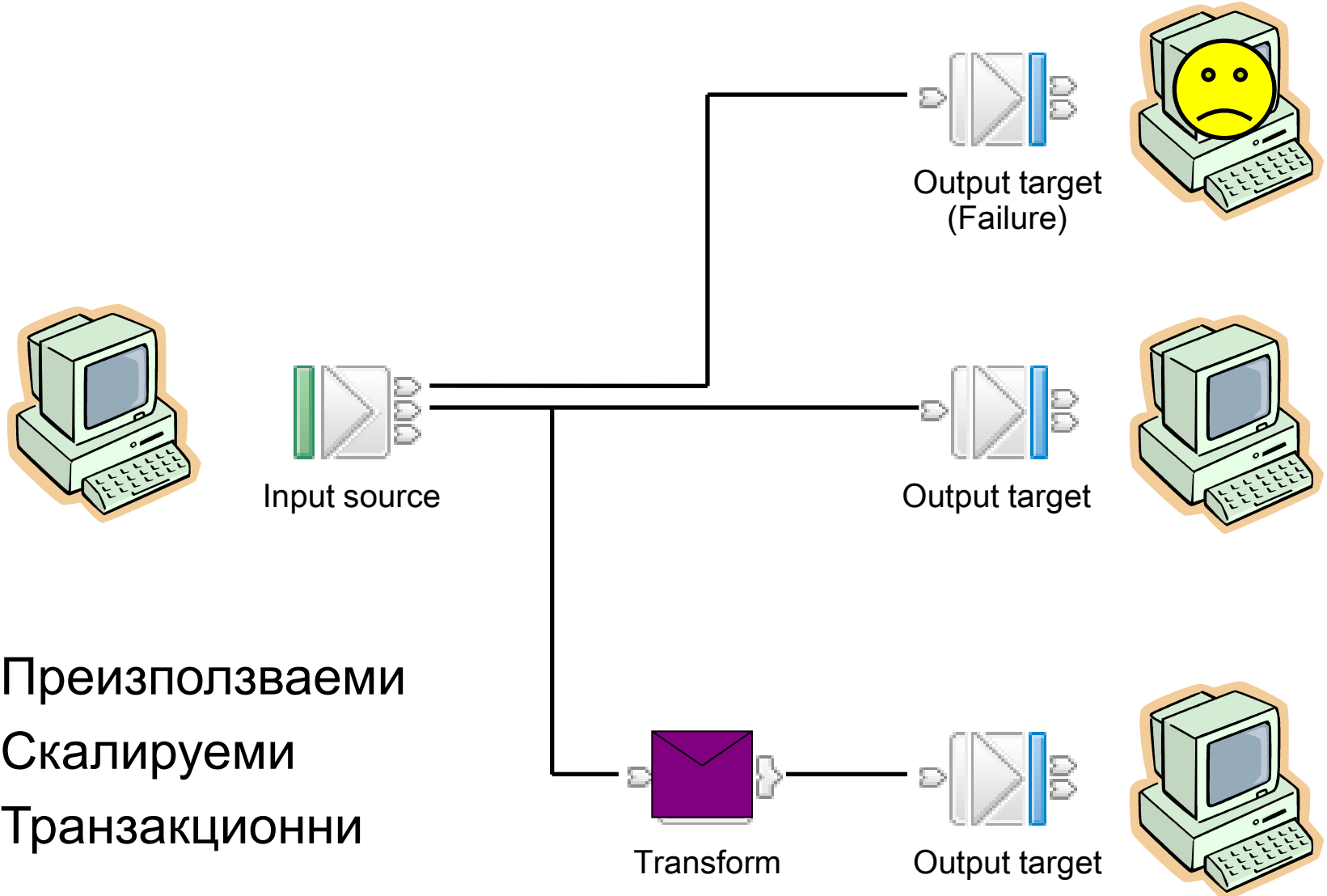
Point-to-point връзките бързо преминават с спагети.

Резултат: Силно обвързана архитектура

- скъпа за поддръжка
- трудна за модификация

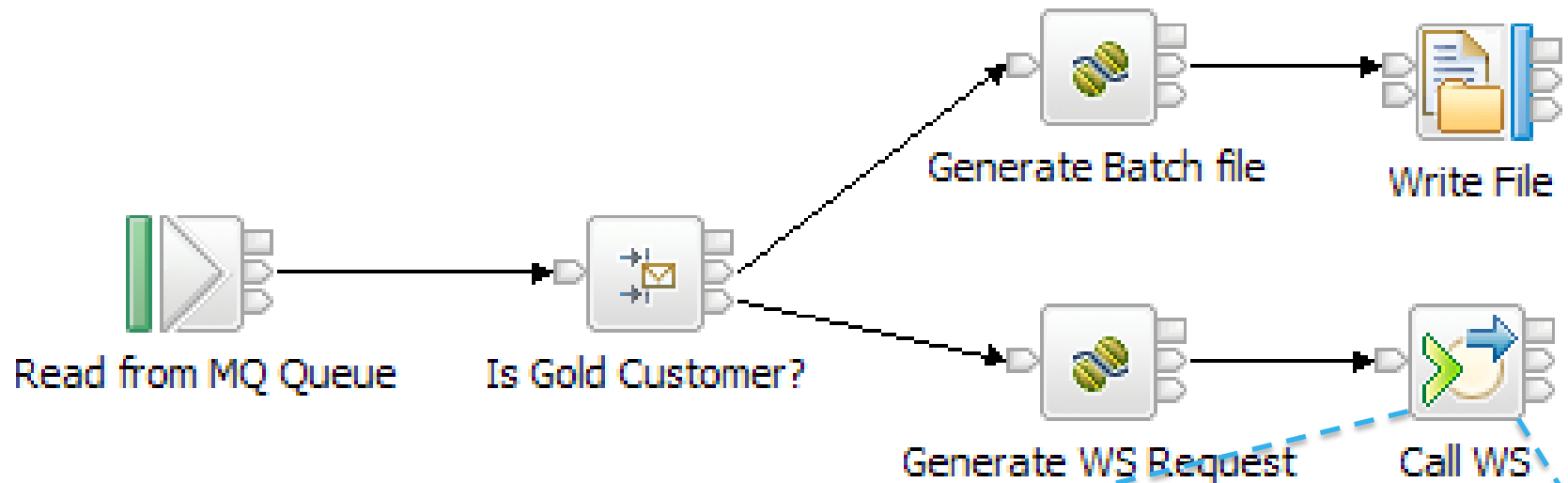


Message flows



- Преизползваеми
- Скалируеми
- Транзакционни

Message flows



HTTP Request Node Properties - HTTP Request

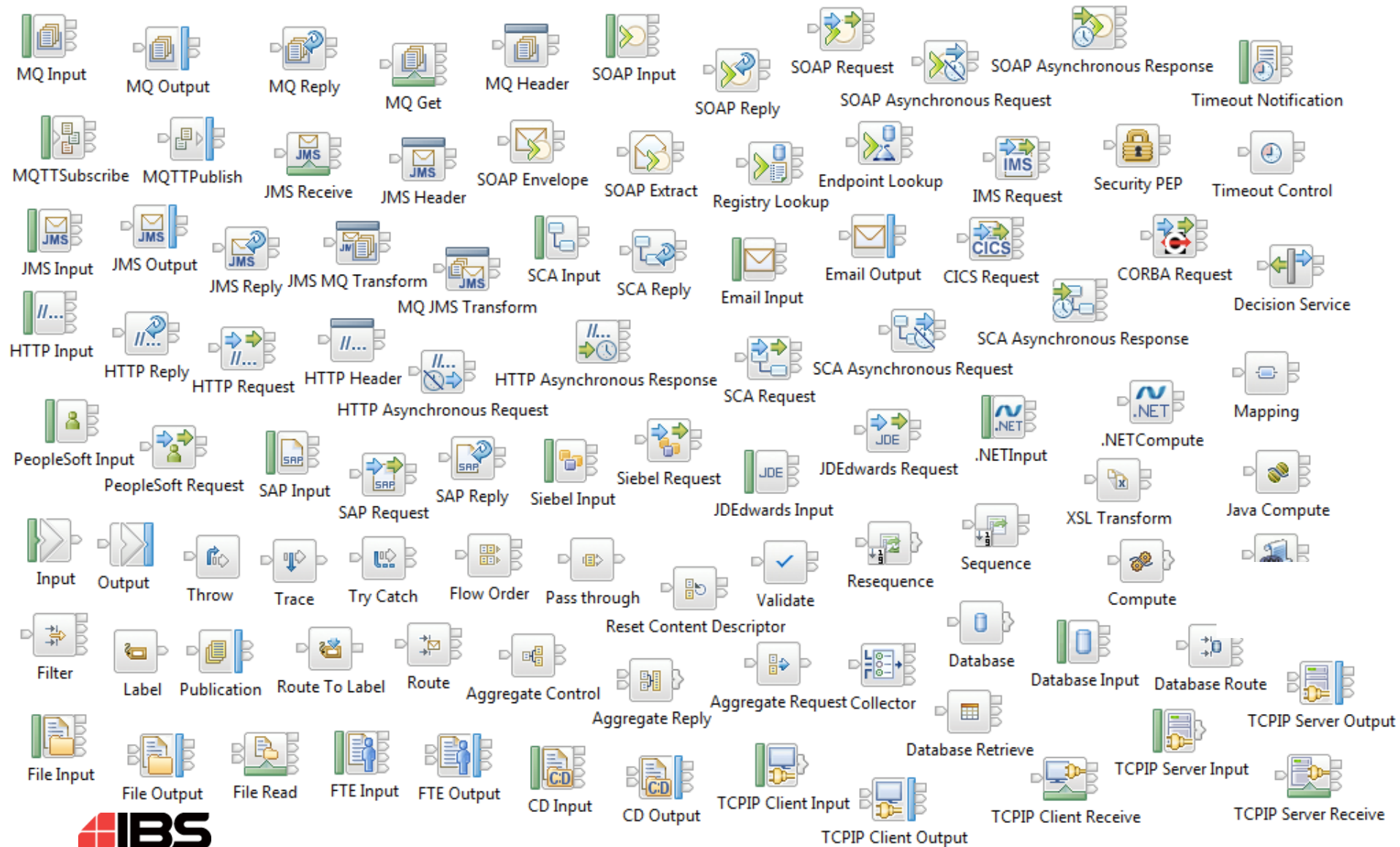
Description	Settings for working with the HTTPRequest node. More...	
Basic	Web service URL*	http://example.org/MyWebService
HTTP Settings		<i>e.g. http://server/path/to/service</i>
SSL	Request timeout (sec)*	120
Response Message Parsing		
Parser Options		

Nodes

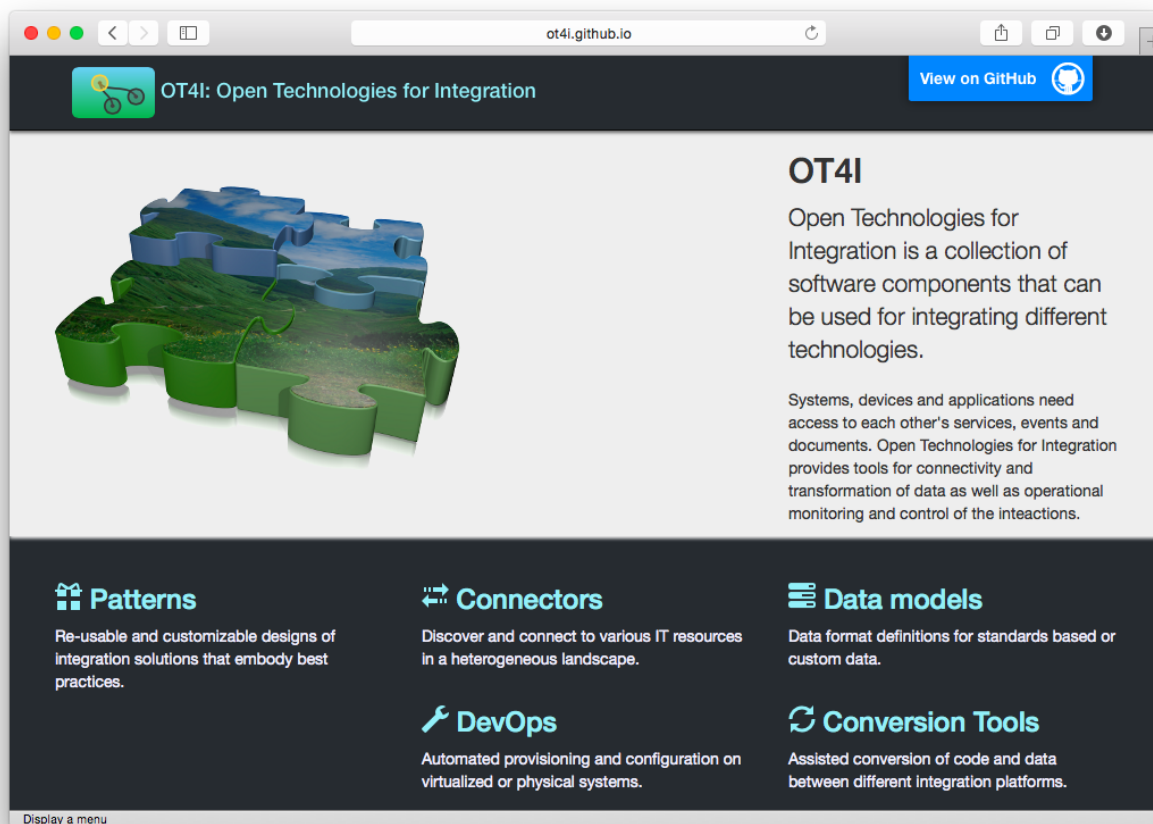
- Основните компоненти на message flow
- Всеки тип node извършва различна операция (вход, изход, обработка, трансформация, ...)
- Предоставени са множество node-ове групирани в различни логически категории



Някои от стандартните nodes

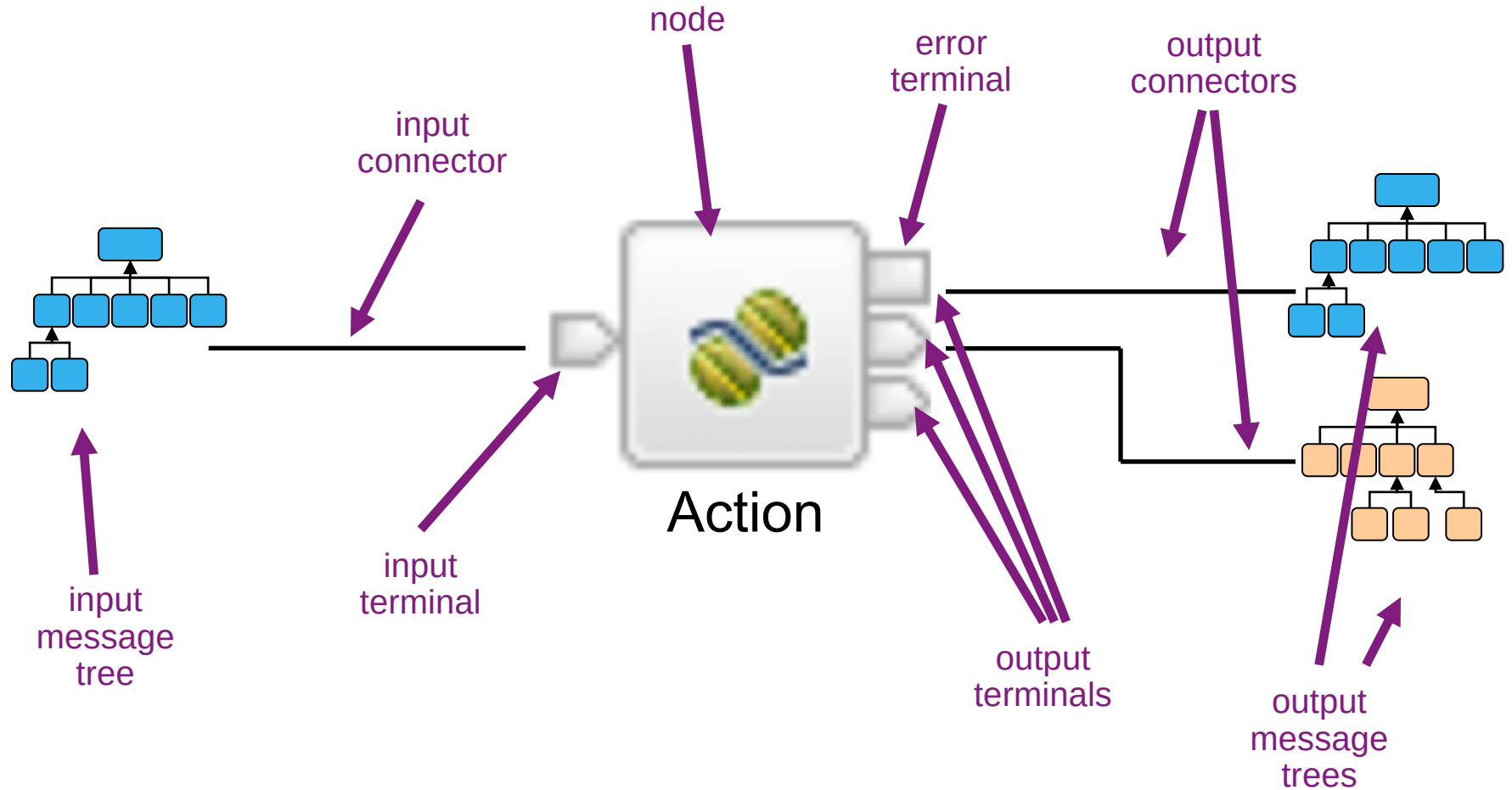


IBM допълнителни компоненти

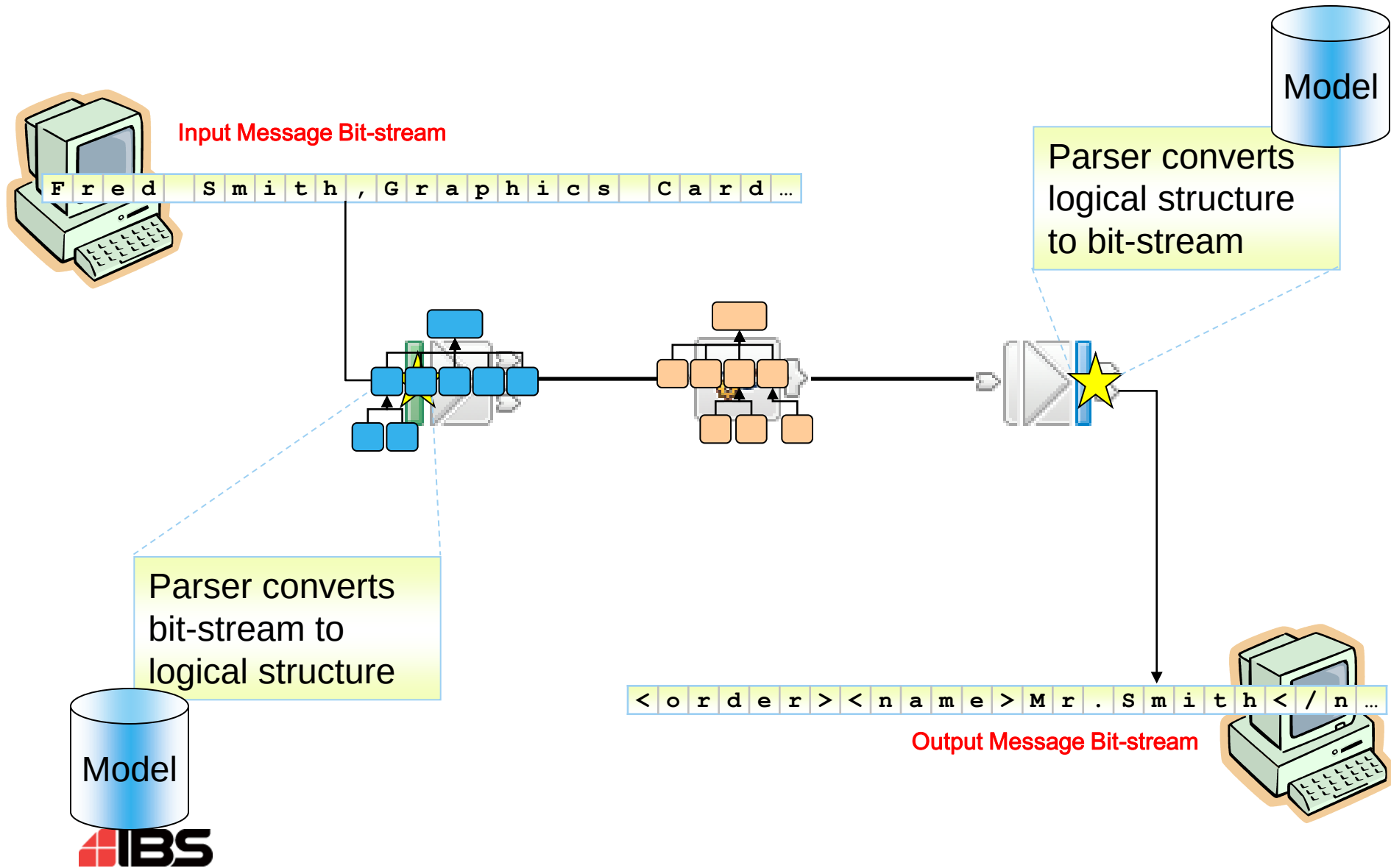


- Съществуват много допълнителни конектори, node-ове и plug-ins
- Писане на собствени User-Defined Node-ове
 - Поддържа се API за това за C и за Java

Node терминология

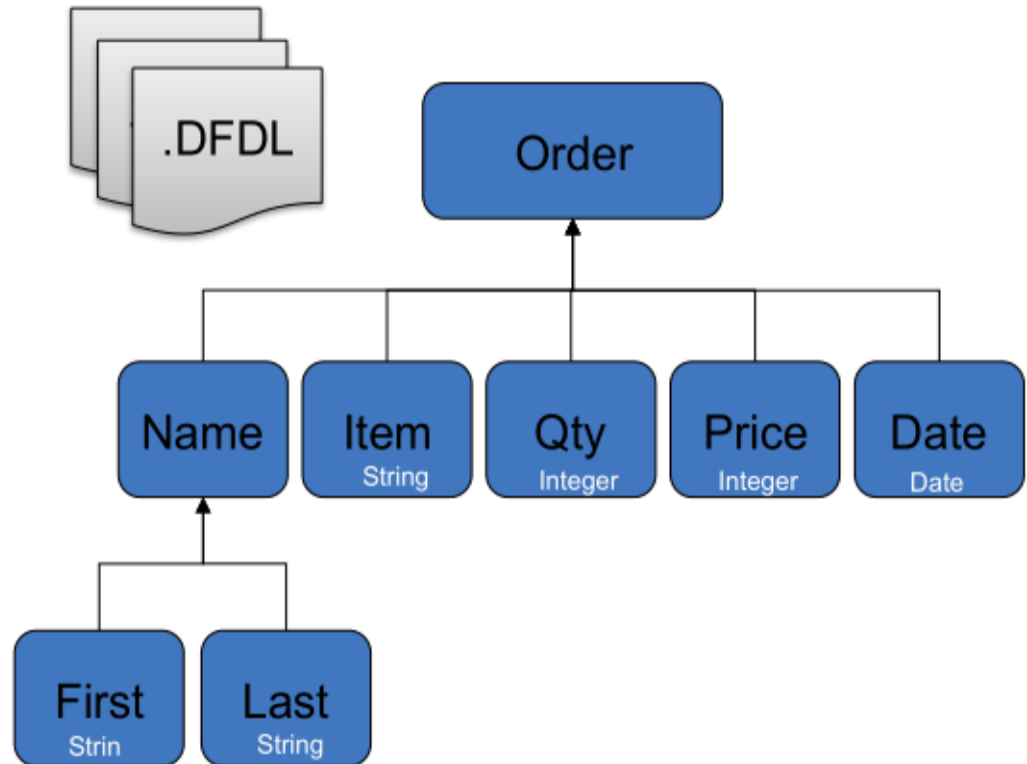
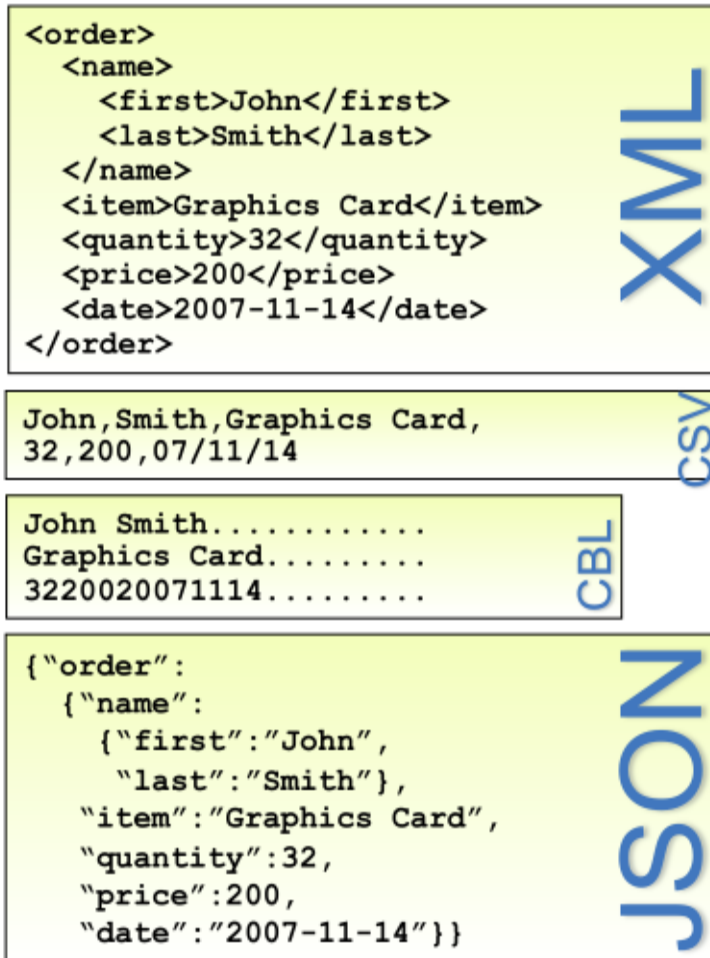


Обработка на данните

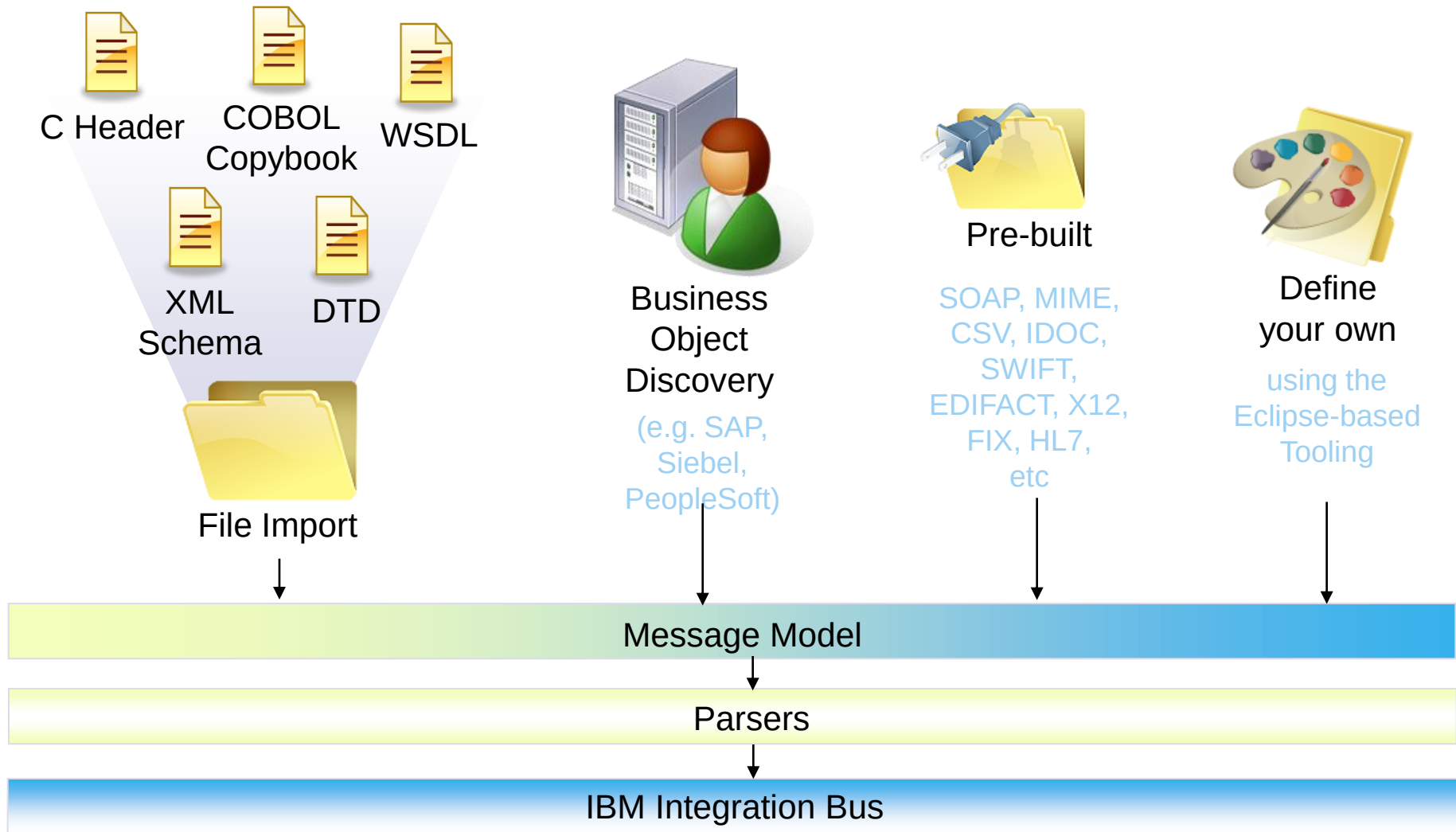


Модел на съобщенията

Physical ← → Logical



Създаване на модели за съобщения



Опции за трансформация на съобщения



Compute

- Describe powerful transformations quickly
- Uses SQL-based language (ESQL)



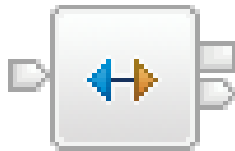
JavaCompute

- Uses Java programming language
- Ability to use XPath



.NETCompute

- Invoke general purpose logic in any .NET supported language
- Windows only



Mapping

- Graphical, easy to use
- Drag and Drop fields, apply functions



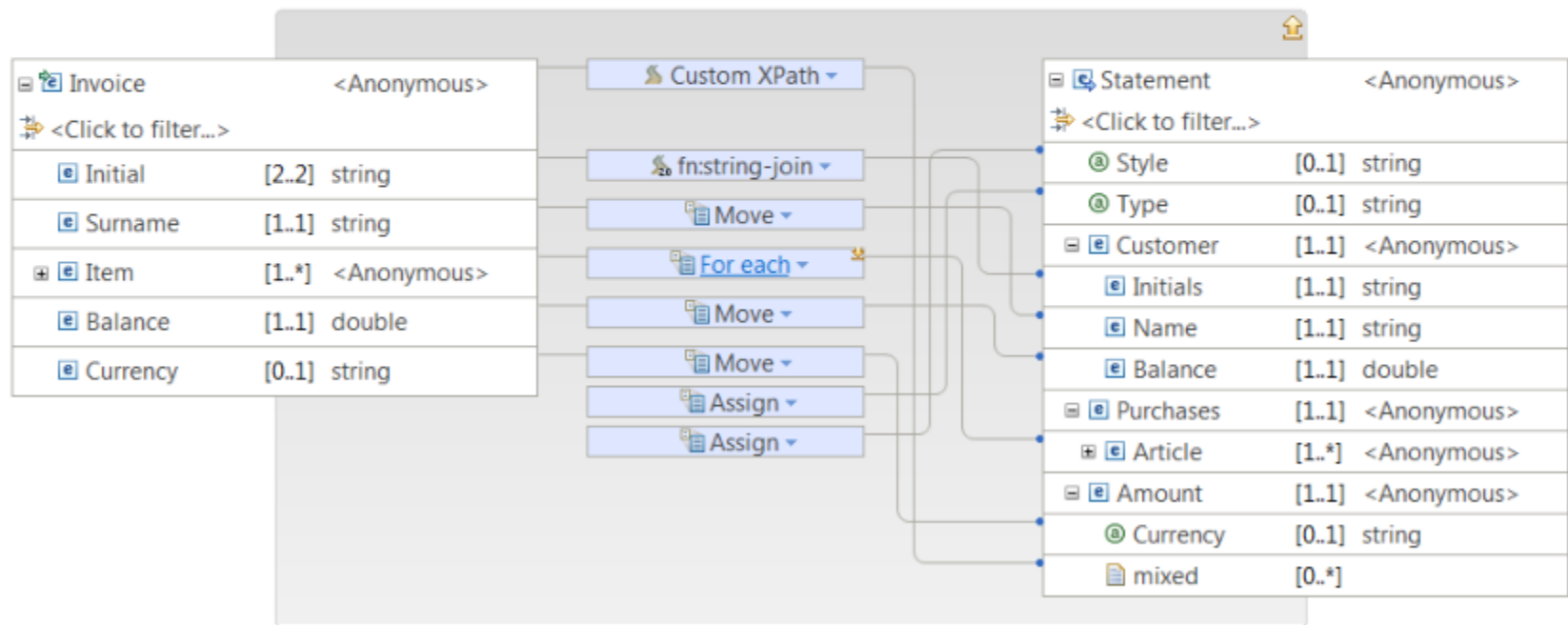
XSL Transform

- Convert XML to anything
- Uses standard XSL Style sheets

Визуален мапинг на елементите



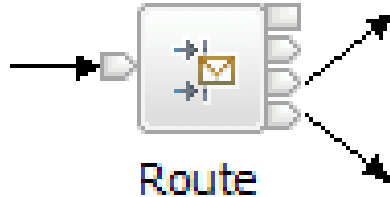
Mapping



Адресиране на елементите на съобщение



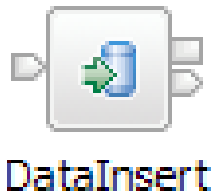
```
public class jcn extends MbJavaComputeNode {  
    public void evaluate(MbMessageAssembly assembly) throws MbException {  
        ...  
        String lastName =  
            (String) assembly.getMessage().evaluateXPath("/Body/Order/Name/Last");  
        ...  
    }  
}
```



Route Node Properties - Route

Filter table*

Filter pattern	Routing output terminal
<code>\$Body/Order/Price > 1000</code>	BigSpenders
<code>\$Body/Order/Price < 200</code>	Cheapskates



```
IF Body.Order.Date < '2008/01/01' THEN  
    INSERT INTO Database.OldOrders (LastName, Item, Quantity)  
    VALUES (Body.Order.Name.Last,  
            Body.Order.Item,  
            Body.Order.Quantity);  
ENDIF;
```

Мониторинг

PurchaseOrder - Business Transaction

Define Monitor

Business Transaction

TotalPurchaseOrderFlow RequestReplyApp CheckAddress

No flow selected

Business Transaction Event Definitions

This section shows the monitoring events that are defined for the selected flow or for all flows in the business transaction. Select the monitoring events that signify the start, transaction.

Message Flow	Event Source Address	Flag as	Event Filter
TotalPurchaseOrderFlow	GoldOrderTotal.terminal.in	Progress	true()
	InputOrder.transaction.End	Progress	\$Root/XMLNSC/PO:purchaseOrder[1]/PO.customerType[1] != 'GUEST'
	InputOrder.transaction.Rollback	Start	\$Root/XMLNSC/PO:purchaseOrder[1]/PO.customerType[1] != 'GUEST'
	InputOrder.transaction.Start	End	\$Root/XMLNSC/PO:purchaseOrder[1]/PO.customerType[1] != 'GUEST'
		Failure	\$Root/XMLNSC/PO:purchaseOrder[1]/PO.customerType[1] != 'GUEST'

Проследяване на бизнес процеси

- визуализация на статуса
- одит лог на преминаващите съобщения

Data Capture Store

Data viewer Replay list

DefaultCaptureStore Mark for replay

	Event time	Local Transaction ID	Parent Transaction
☒	2012-01-24 16:16:48.716	TESTDATA0123456789@!	CDA



Мониторинг



Data Capture Store

Data viewer

Replay list

eikpodMonitoring

☒ Mark for replay

Customize

20 of 29078 items shown [Clear Filter](#)

	Време	MPH	Parent Transaction ID	Състояние	Данни	Errors	Действие
☒	8.10.2016 г. 13:36:08.915 GMT+3	16TR35340000011489	10.30.84.18	BG001015		-	New EAD
☐	8.10.2016 г. 13:36:21.511 GMT+3	16TR35340000011489	BG001015	SUCCESS		-	Send EAD to portal
☐	9.10.2016 г. 12:28:44.383 GMT+3	16TR35340000011489	CONTROL_REQ UIRED			-	Portal EAD Report
☐	9.10.2016 г. 12:28:50.716 GMT+3	16TR35340000011489	BG001015			-	CCN Request Service
☐	9.10.2016 г. 12:28:57.30 GMT+3	16TR35340000011489		1		-	CCN Callback Service

Поддръжка на SWAGGER



```
2 # as a demonstration of an API spec in YAML
3 swagger: '2.0'
4 info:
5   title: Uber API
6   description: Move your app forward with the Uber API
7   version: "1.0.0"
8   # the domain of the service
9   host: api.uber.com
10  # array of all schemes that your API supports
11  schemes:
12    - https
13  # will be prefixed to all paths
14  basePath: /v1
15  produces:
16    - application/json
17  paths:
18    /products:
19      get:
20        summary: Product Types
21        description: |
22          The Products endpoint returns information about the "Uber" products
23          offered at a given location. The response includes the display name
24          and other details about each product, and lists the products in the
25          proper display order.
26        parameters:
27          - name: latitude
28            in: query
29            description: Latitude component of location.
30            required: true
31            type: number
32            format: double
33          - name: longitude
34            in: query
35            description: Longitude component of location.
36            required: true
37            type: number
38            format: double
39        tags:
40          - Products
41        responses:
42          200:
43            description: An array of products
44            schema:
45              type: array
46              items:
47                $ref: '#/definitions/Product'
48        default:
```

Paths

/products

GET /products

Products

Summary

Product Types

Description

The Products endpoint returns information about the *Uber* products offered at a given location. The response includes the display name and other details about each product, and lists the products in the proper display order.

Parameters

Name	Located in	Description	Required	Schema
latitude	query	Latitude component of location.	Yes	⇌ number (double)
longitude	query	Longitude component of location.	Yes	⇌ number (double)

Responses

Code	Description	Schema
200	An array of products	⇌ ▼ Product { }
default	Unexpected error	⇌ ▼ Error { code: integer message: string fields: string }

Try this operation

REST API поддръжка

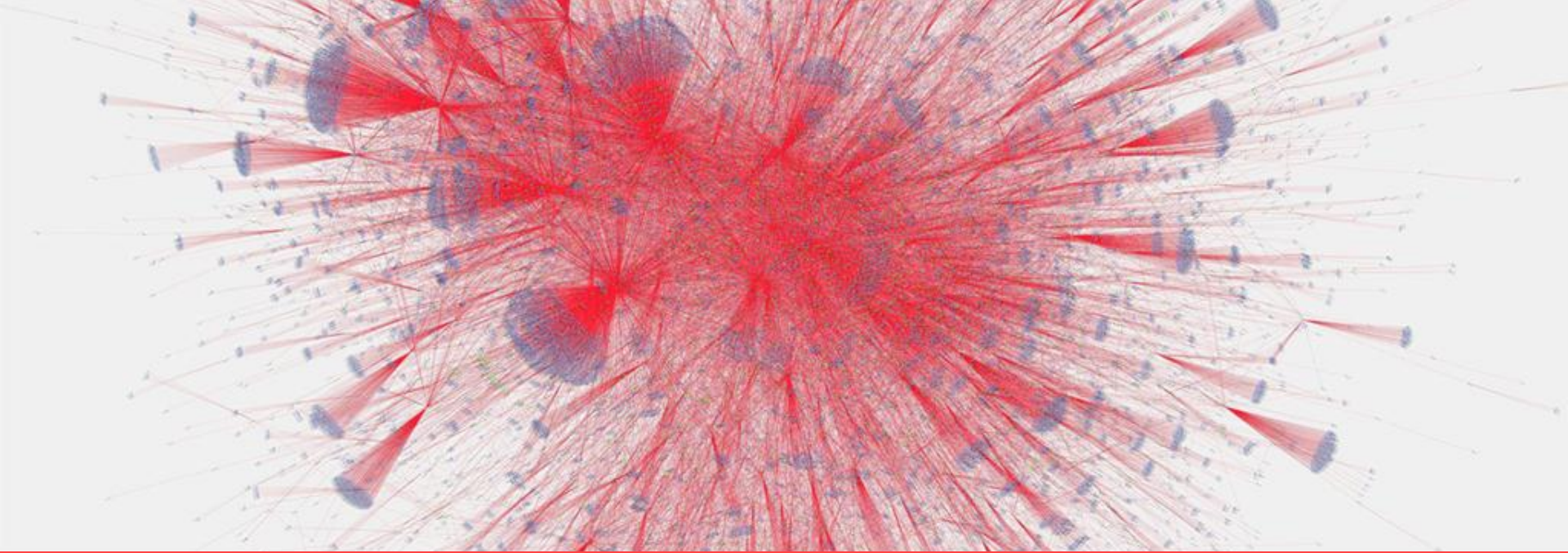
Header

The screenshot displays the IBM Integration Center (IIB) console interface. The browser address bar shows the URL: `localhost:4418/#restApi/1/executiongroups/default/restapis/CustomerDatabaseV1`. The left sidebar shows a tree view of the system configuration, with **CustomerDatabaseV1** selected under **REST APIs**. The main panel displays the **CustomerDatabaseV1 - REST API** configuration page, which includes tabs for **Overview**, **API**, and **Statistics**. The **API** tab is active, showing the following details:

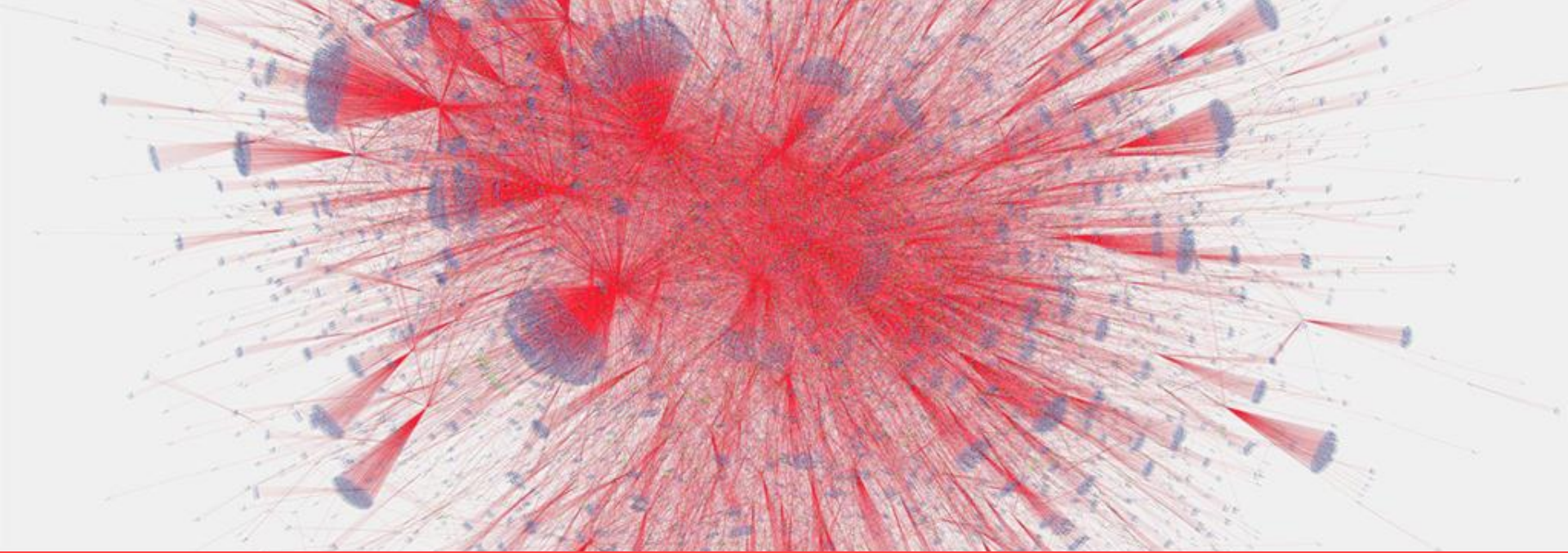
- Base URL For Remote Invocations:** `http://9.140.102.163:7800/customerdbv1`
- Remote URL For The REST API Definitions:** `http://9.140.102.163:7800/customerdbv1/swagger.json`
- Base URL For Local Invocations:** `http://localhost:7800/customerdbv1`
- Local URL For The REST API Definitions:** `http://localhost:7800/customerdbv1/swagger.json`

The **API** tab also lists the following endpoints and their status:

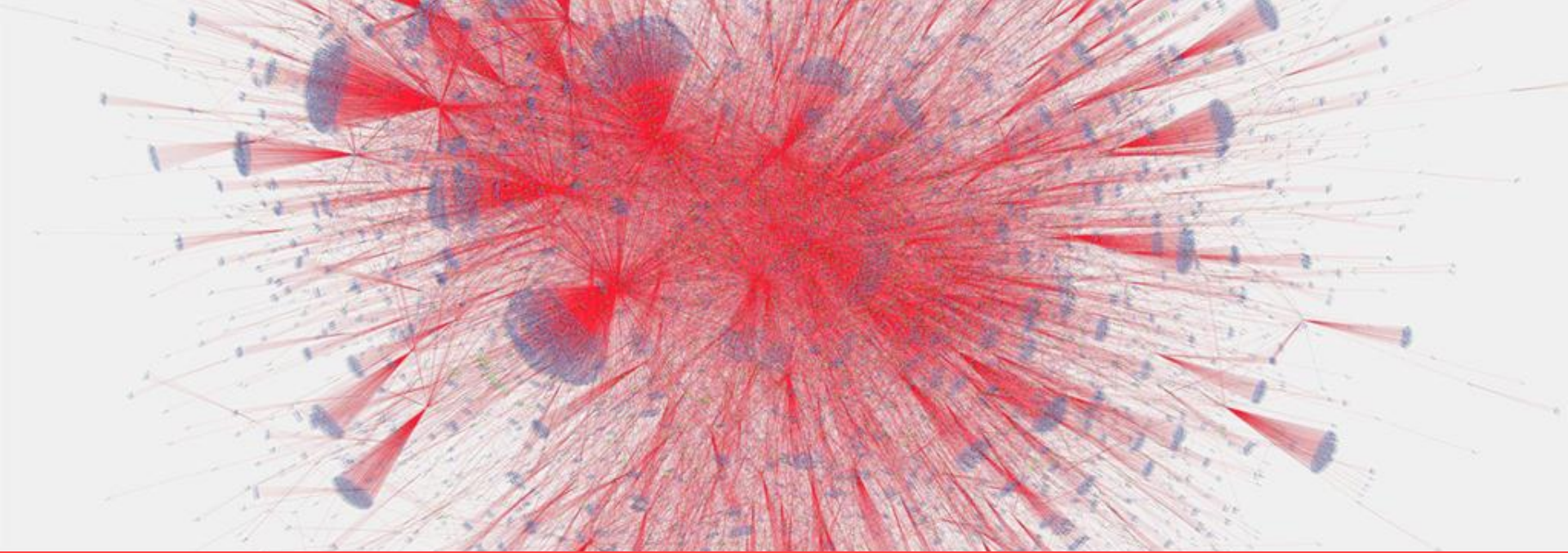
Method	Endpoint	Description	Status
POST	<code>addCustomer</code>	Add a customer to the database	Implemented
GET	<code>getAllCustomers</code>	Get all customers from the database	Implemented
DELETE	<code>deleteCustomer</code>	Delete a specified customer from the database	Implemented
GET	<code>getCustomer</code>	Get a specified customer from the database	Implemented
PUT	<code>updateCustomer</code>	Update a customer in the database	Implemented



DEMO



THE END



Q & A