Mainframe Integration – Handling Legacy Application Challenges

Glenn Schneck

Principal Techincal Architect -
GT Software

gschneck@gtsoftware.com

GTSoftware®

IBM Mainframes Retain the System of Record

CICS Systems for Banks
(ATM, loans, account management)

CICS Systems for Insurance
(Claims & policy management)

Mainframe Systems for Manufacturing

Mainframe Systems for Finance

Mainframe Systems for Medical

Mainframe Systems for Airline

GTSoftware®

Mainframes are DEAD??

- 71% of Fortune 500 companies use mainframes
- Mainframes handle 87% of all credit card transactions
- Mainframes handle 68% of the world's production IT resources, yet they account for only 6% of IT costs
- Mainframes handle 30 BILLION business transactions a day
- As of 2017, 92 of the world's top 100 Banks continued to use mainframes

Source: https://blog.syncsort.com/2018/06/mainframe/9-mainframe-statistics/

GTSoftware®

Today's Business Needs

Build and deploy API's rapidly
Web self-service, mobile/cloud, BYOD
Real-time access to enterprise data residing on any platform
Integrated views of related information
Customer and business focused IT
Industry standards
Integration between mainframe & distributed systems

GTSoftware®

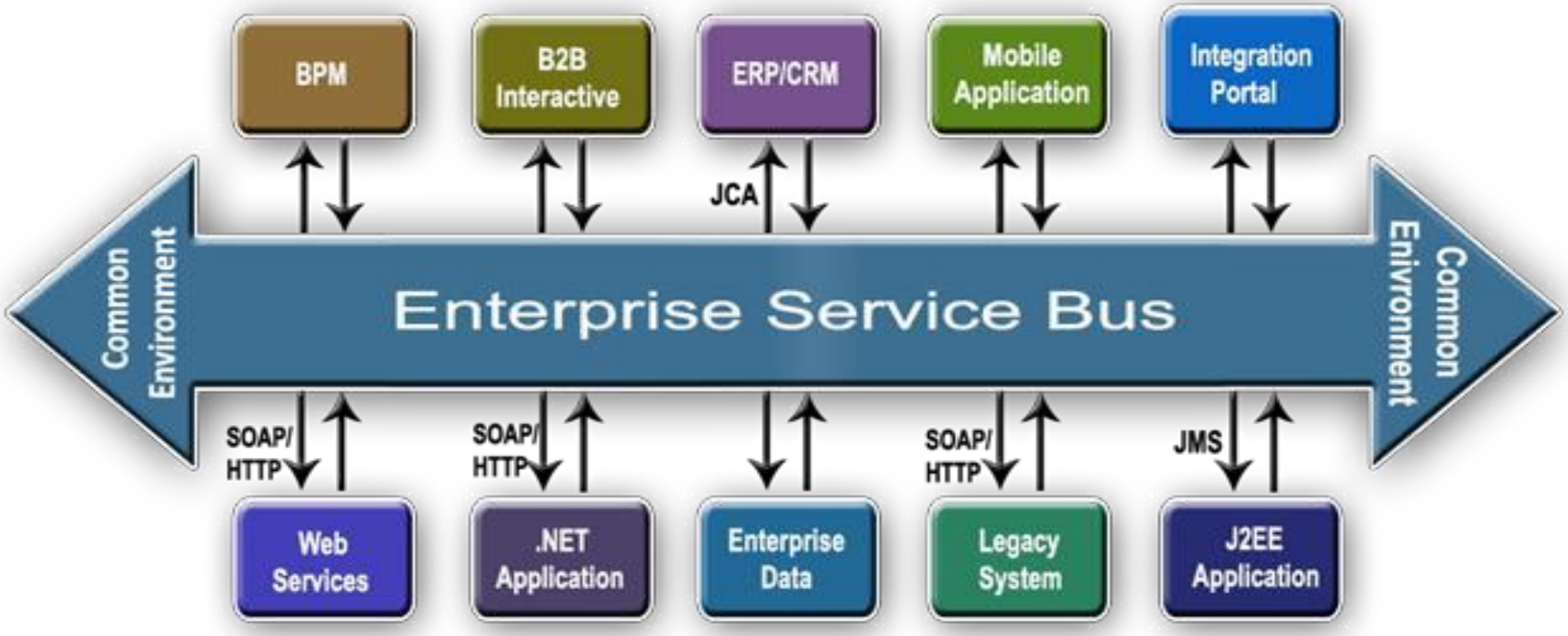
***CICS API's** : We are already doing that!!!!!*

Lessons Learned, War Stories, Successes

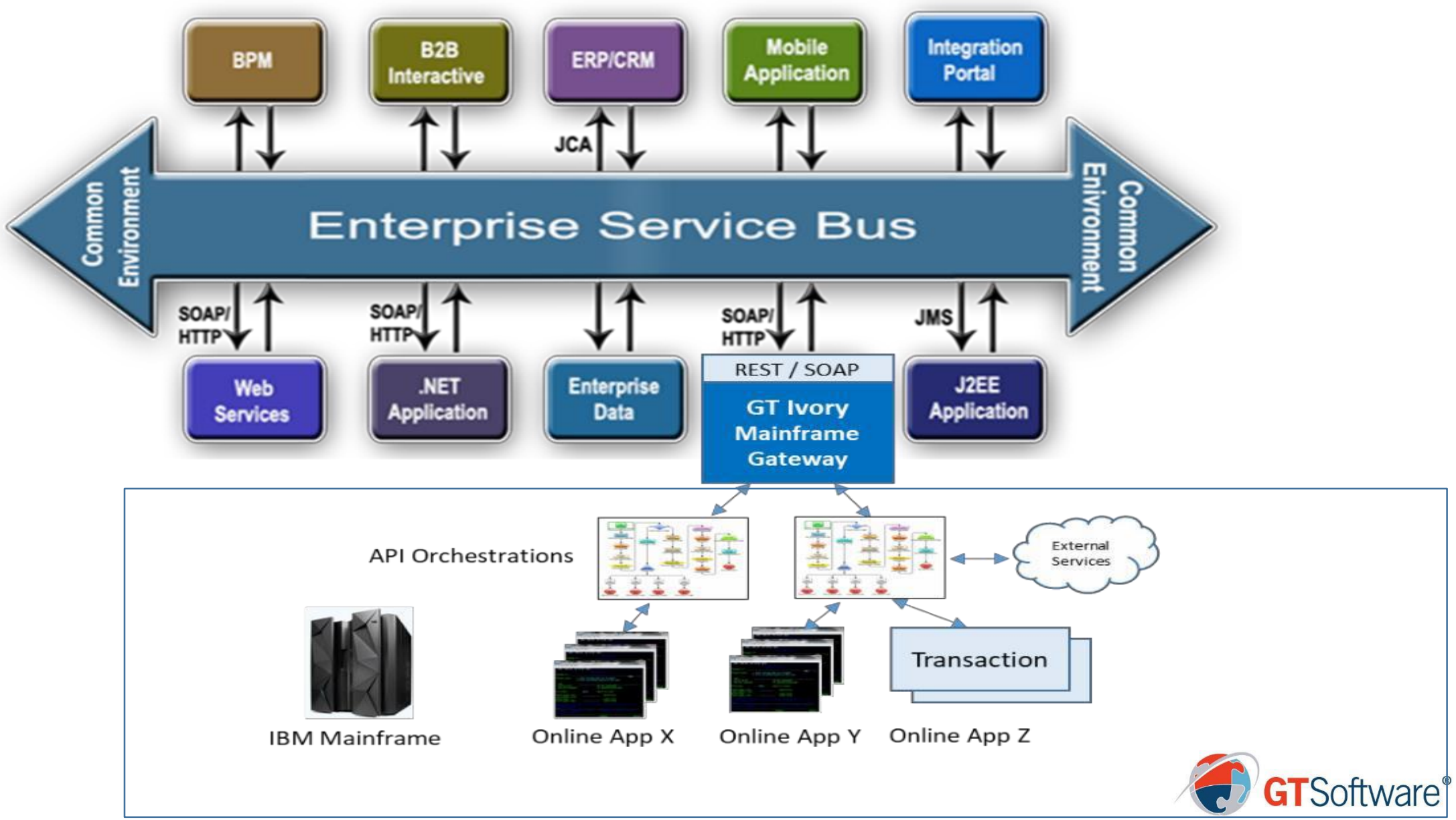
GTSoftware®

We have an ESB to do all that!!!!!!!!!!!!!!!

GTSoftware®


BPM
B2B Interactive
ERP/CRM
Mobile Application
Integration Portal
JCA
Common Environment
Enterprise Service Bus
Common Enivronment
SOAP/ HTTP
SOAP/ HTTP
SOAP/ HTTP
JMS
Web Services
.NET Application
Enterprise Data
Legacy System
J2EE Application

GTSoftware®


BPM
B2B Interactive
ERP/CRM
Mobile Application
Integration Portal
JCA
Common Environment
Enterprise Service Bus
Common Enivronment
SOAP/ HTTP
SOAP/ HTTP
SOAP/ HTTP
JMS
Web Services
.NET Application
Enterprise Data
REST / SOAP
GT Ivory Mainframe Gateway
J2EE Application
API Orchestrations
External Services
Transaction
IBM Mainframe
Online App X
Online App Y
Online App Z
GTSoftware®

What else do we need to consider??

GTSoftware®

Do You Have the Right Mainframe Integration Technologies?

- How old are your legacy backend applications?
- What technologies are they using?
- Is the application code structured or unstructured?
- Did your core applications first start out as commercial offerings?
- What third-party components are embedded in the code?
- How complex is the code and data structure design?
- Do your support teams fully understand the application?
- How many coding ‘standards’ have been used over the past years?

GTSoftware®

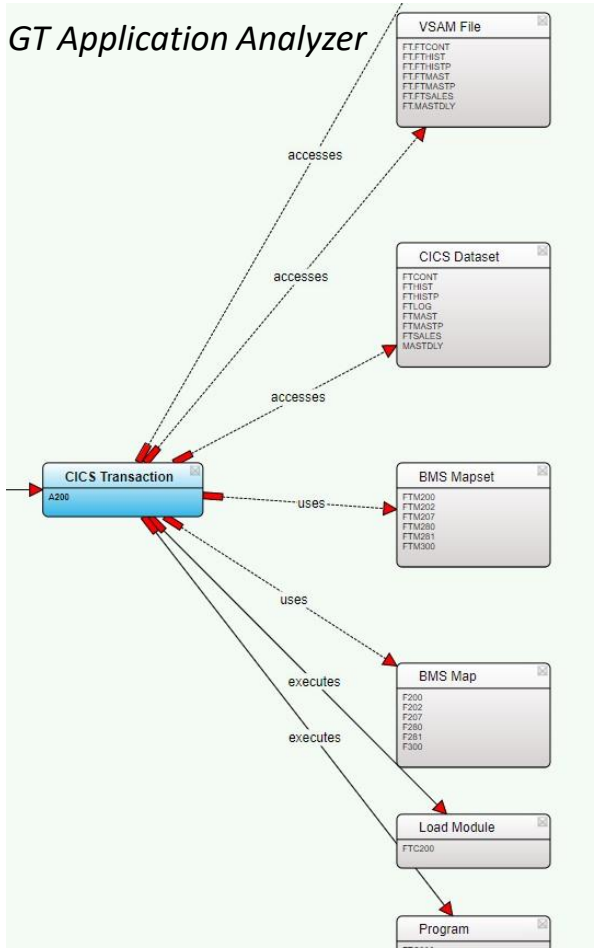
Understanding Your Legacy Applications & API Requirements

- Most mainframe online applications were designed to interact with 3270 terminals (end user dialog).
- Integration technologies should be transparent to the backend systems.
- Changing legacy code to work better as an API introduces more complexity and code to manage.
- Fine grain APIs (microservices) may be easier to build, but put more work onto the consumer.
- More intelligent the API, less effort for the API consumer processing logic.
- Legacy mainframe apps are like a box of chocolates, it is hard to see what is inside.

GTSoftware®

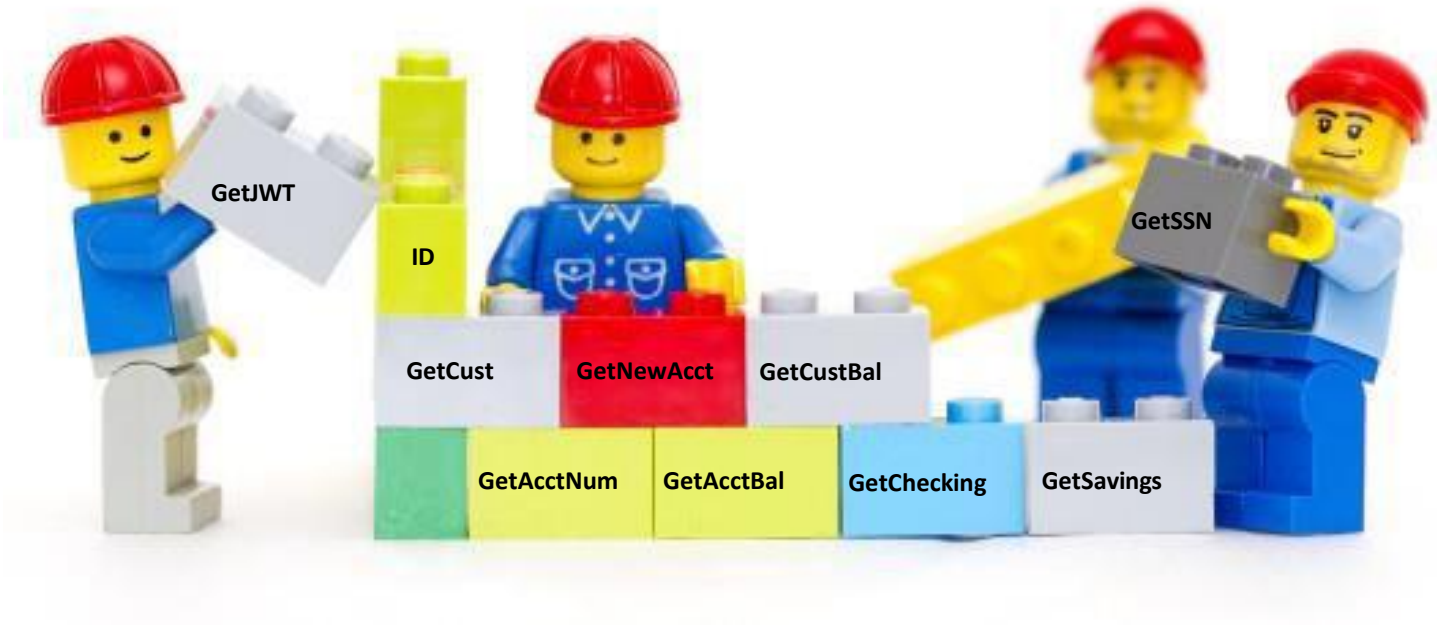
Legacy Application Complexities – CICS and IMS

- XCTL/ multiple program calls
- Multiple input and output messages
- Variable length, multi-part messages, different layouts
- Complex structures (REDEFINES & ODO)
- Null terminations, non-standard code
- Screen macros
- Conversational dialogs
- External and other 3270 applications
- Complex Conversational Transactions



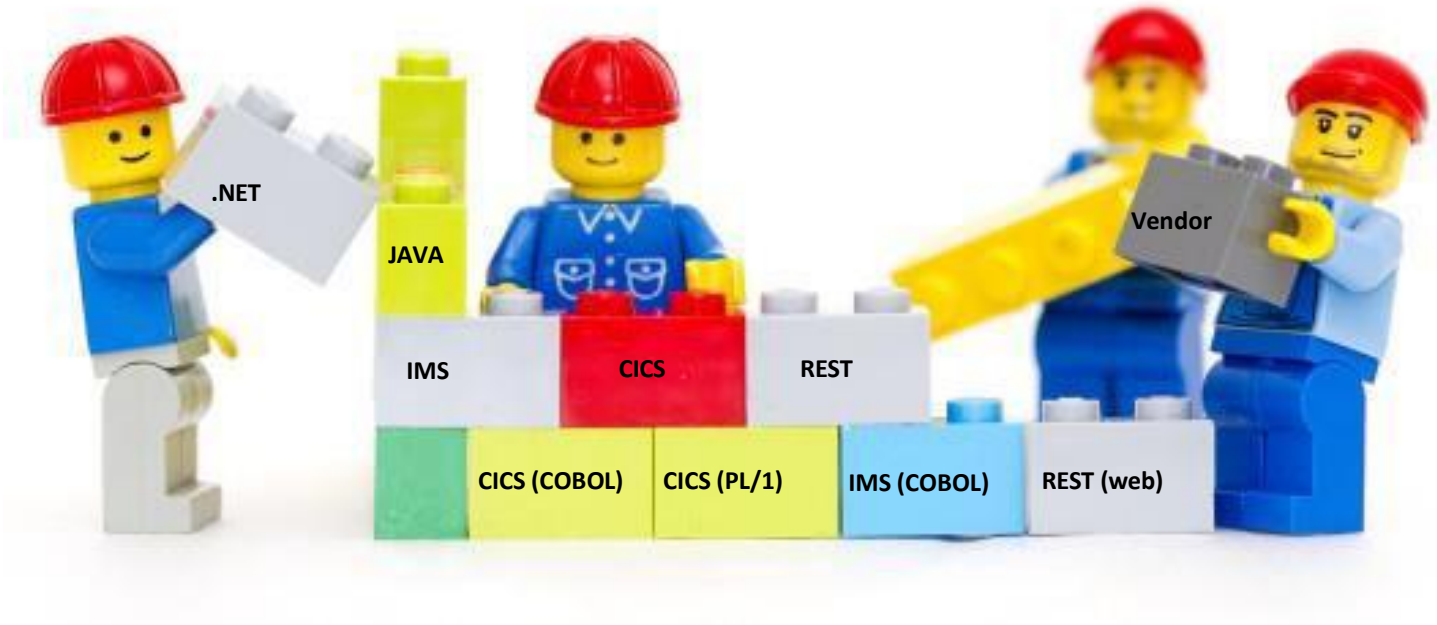
GTSoftware®

APIs are building blocks



GTSoftware®

APIs are building blocks



GTSoftware®

Mainframe APIs – Understanding Legacy Environments & Integration Requirements

Web
Mobile
COTS
ESB
Cloud
BI Tools
Blockchain
AI
RPA
Other

Consumer API Processing Logic

API Manager

API requests and replies focused a single transaction (microservice)… GetChecking, GetSavings, GetCarLoan

Transaction Message API

Microservice

REST API

input JSON

output JSON

mapping

input msg

output msg

Provider

(1 input and 1 output)

- CICS Program
- IMS Program
- MQ
- Batch
- DB2 (SQL)
- IMS DB (SQL)

GTSoftware®

Mainframe APIs – Understanding Legacy Environments & Integration Requirements

LOBs

Web
Mobile
COTS
ESB
Cloud
BI Tools
Blockchain
AI
RPA
Other

Consumer API Processing Logic

API Manager

API requests and replies focused a single transaction (microservice)… GetChecking, GetSavings, GetCarLoan

Transaction Message API

z/OS Connect EE

REST API

input JSON

mapping

input msg

msg

Simple Rest API Provider

Provider

(1 input and 1 output)

- CICS Program
- IMS Program
- MQ
- Batch
- DB2 (SQL)
- IMS DB (SQL)

- Data manipulation (math, string)
- Flow control (if, then)

input msg
output msg
output msg

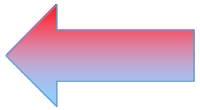
Complex Transactions (CICS & IMS)

- Message switching / program calls
- Multiple input and output messages
- Variable length, multi-part messages
- Complex structures (REDEFINES & ODO)
- Null terminations, non-standard code

- Multiple transactions (logical sequence)

- Screen macros (BMS and MFS logic)
- Third-party, remote or other 3270 apps (TN3270, IDMS, NATURAL, other)
- Conversational dialogs

Each consumer must handle



GTSoftware®

Mainframe APIs – Understanding Legacy Environments & Integration Requirements

Web
Mobile
COTS
ESB
Cloud
BI Tools
Blockchain
AI
RPA
Other

Consumer API Processing Logic

API Manager

API requests and replies focused a single transaction (microservice)… GetChecking, GetSavings, GetCarLoan

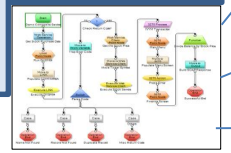
API requests and replies focused on a logical process (composite)… GetCustomer

Business Process Workflow

REST API

SOAP API

GT Ivory Smart API



No coding

External Services

- Distributed apps and data (DVM)
- Cloud services
- Other API (REST, SOAP, other)

Transaction Message API

Microservice

REST API | input JSON / output JSON | mapping | input msg / output msg | Provider

(1 input and 1 output)

- CICS Program
- IMS Program
- MQ
- Batch
- DB2 (SQL)
- IMS DB (SQL)

- Data manipulation (math, string)
- Flow control (if, then)

input msg
output msg
output msg

Complex Transactions (CICS & IMS)

- Message switching / program calls
- Multiple input and output messages
- Variable length, multi-part messages
- Complex structures (REDEFINES & ODO)
- Null terminations, non-standard code

- Multiple transactions (logical sequence)

- Screen macros (BMS and MFS logic)
- Third-party, remote or other 3270 apps (TN3270, IDMS, NATURAL, other)
- Conversational dialogs

GT Ivory Runtime Options – z/OS, IFL (Linux on z), Linux Server, Windows/Java Server, Hybrid

GTSoftware®

Environment

- CICS

GTSoftware®

Environment

- CICS
- Ivory Service Architect (API creation & orchestration)
- Github (source version control)
- Jenkins (*automation*)
- .NET , Java, Node.js , COBOL
- Linux (redhat) JBOSS
- Tomcat
- MoogSoft
- dynatrace
- urbancode

API Lessons Learned

COBOL and PL/1

THE GOOD	THE BAD	THE UGLY
All Data Structures Supported	Some structures don't map well to distributed Apps	Comp-3, Binary , ODO REDEFINES, unbounded sequences (PL/1)
All can be exposed as service inputs/outputs	Names in COBOL-PL/1 may be cryptic and need to be renamed	Blank When Zero.
Can expose existing programs without changes	May need more data to drive than the app knows	Message switches, and other calls

GTSoftware®

```
01 VAR-RECORD.
 05  REC-OTHER-DATA     PIC X(30).
 05  REC-AMT-CNT        PIC 9(4).
 05  REC-AMT            PIC 9(5)
       OCCURS 1 TO 100 TIMES
            DEPENDING ON REC-AMT-CNT.
  .
```

```
1 INSTRING UNALIGNED,
2 FIX_PART,
3 CERTNO CHAR(9),
3 COUNTZ FIXED DECIMAL(1,0),
2 VAR_PART (7 REFER (COUNTZ)) CHAR(10);
```

GTSoftware®

CICS or IMS Transactions

THE GOOD	THE BAD	THE UGLY
Existing Transactions can be exposed as REST or SOAP	A Transaction may be too fine grained	Multiple Transactions may have to be used in service
Data from transaction returned as a service output	Data may be to convoluted to use in service	Volume of data may be too large to return to distributed client
PFKEY = TRANCODE	Maybe need multiple Trans	Maybe need to call multiple Trans in sequence

GTSoftware®

CICS and/or IMS Transactions Combined

THE GOOD	THE BAD	THE UGLY
Combine Transactions in one service	May not work well with others	API's that run for minutes
Use Conversational Transactions	Long running conversations may be long running API's	No understanding of conversational impact
No Code re-write	May be easier to combine logic to keep from calling multiples transactions	May return different copybook

GTSoftware®

Even Uglier…

Null Termination x’3F’ (IMS MFS)

Ex. 03 LAST-NAME PIC X(20). | ‘RIVERS ‘ D9C9E5C5D9E2404040404040404040404040

To: ‘RIVERS ‘ D9C9E5C5D9E23F’

XML: <lastName>RIVERS3#A2<lastName>

Null Termination x’00’

Ex. 03 NAME PIC X(20). | ‘RIVERS DUSTY ‘ D9C9E5C5D9E2404040000C4D9E2E3E8

To: ‘RIVERS ‘ D9C9E5C5D9E2’

XML: <NAME>RIVERS<NAME>

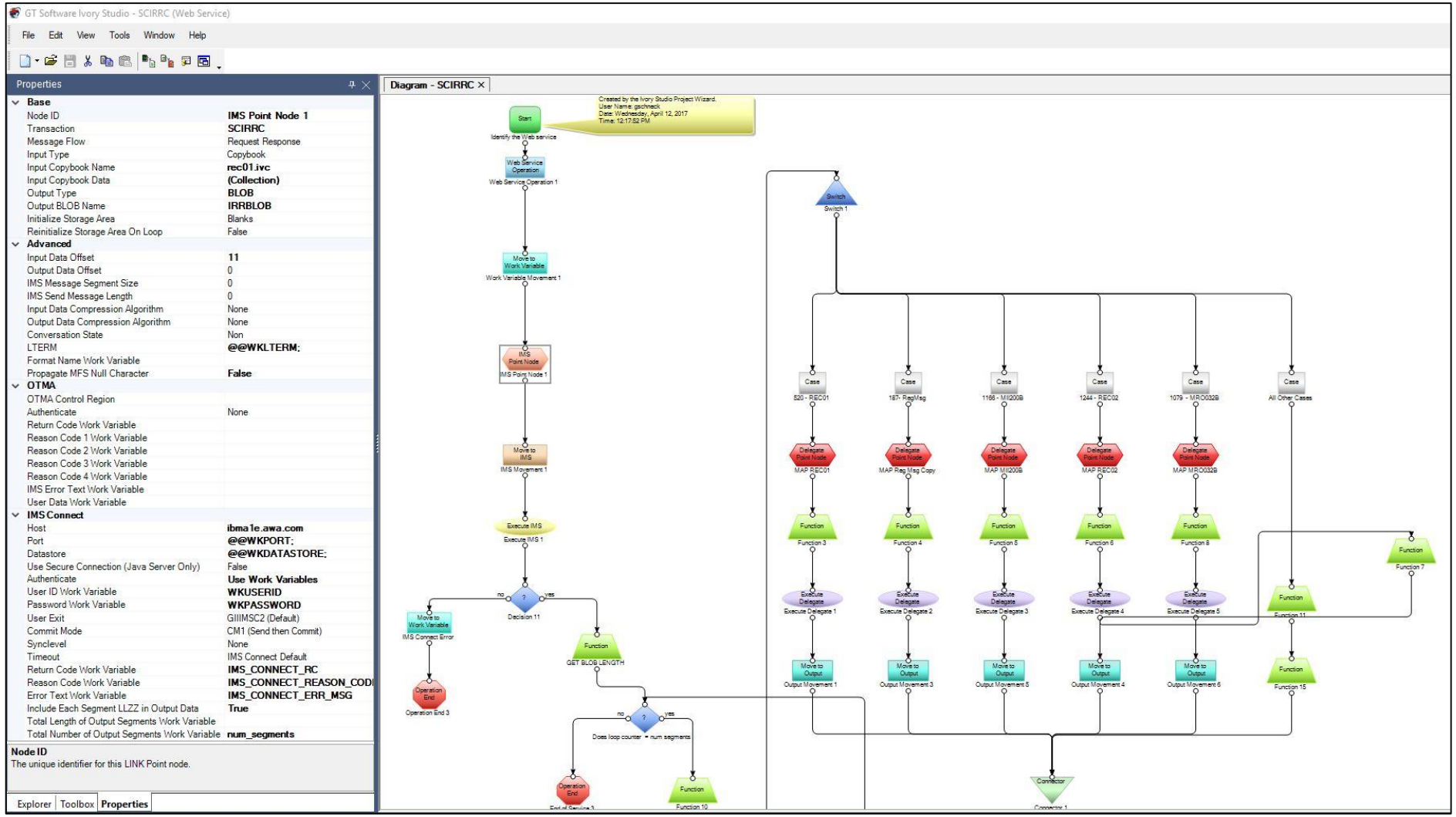
GTSoftware®

Design Methodology

- **Base Services** (closely matched to individual Transactions when possible)
- **Composite Services** (combined calling of multiple base services for business services)
- Outbound calls to third party software from COBOL

GTSoftware®

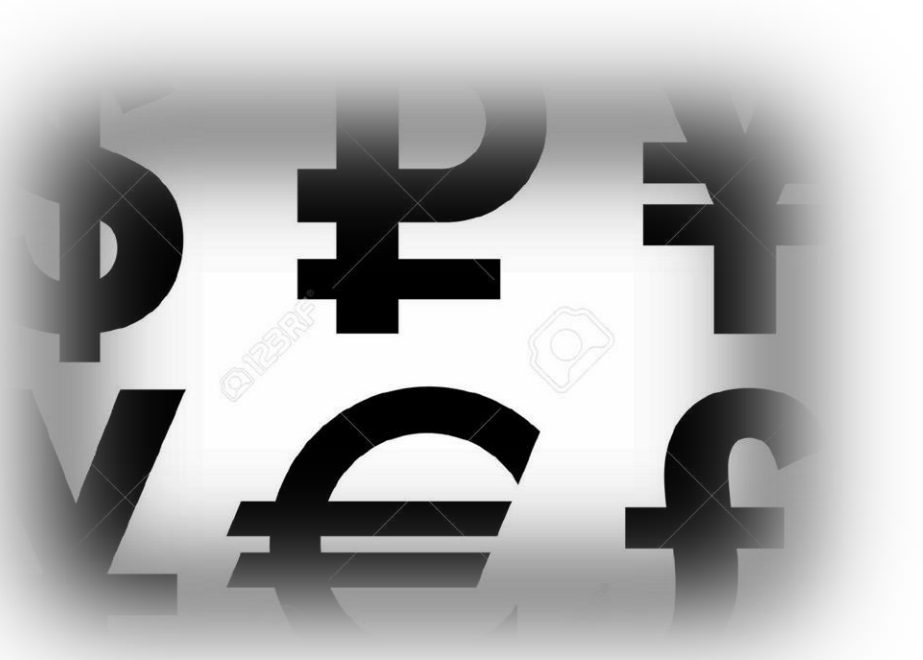
Ivory API – Online Transaction with Multiple Variable-length Output Messages



GTSoftware®

Financial

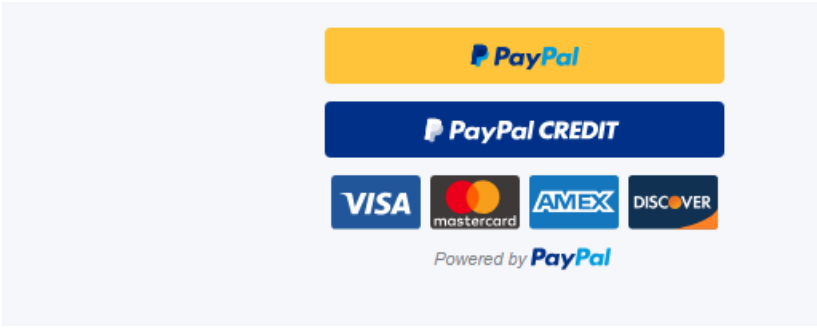
- Domestic Banks
- Domestic Insurance
- International Banks
- International Insurance



Banking

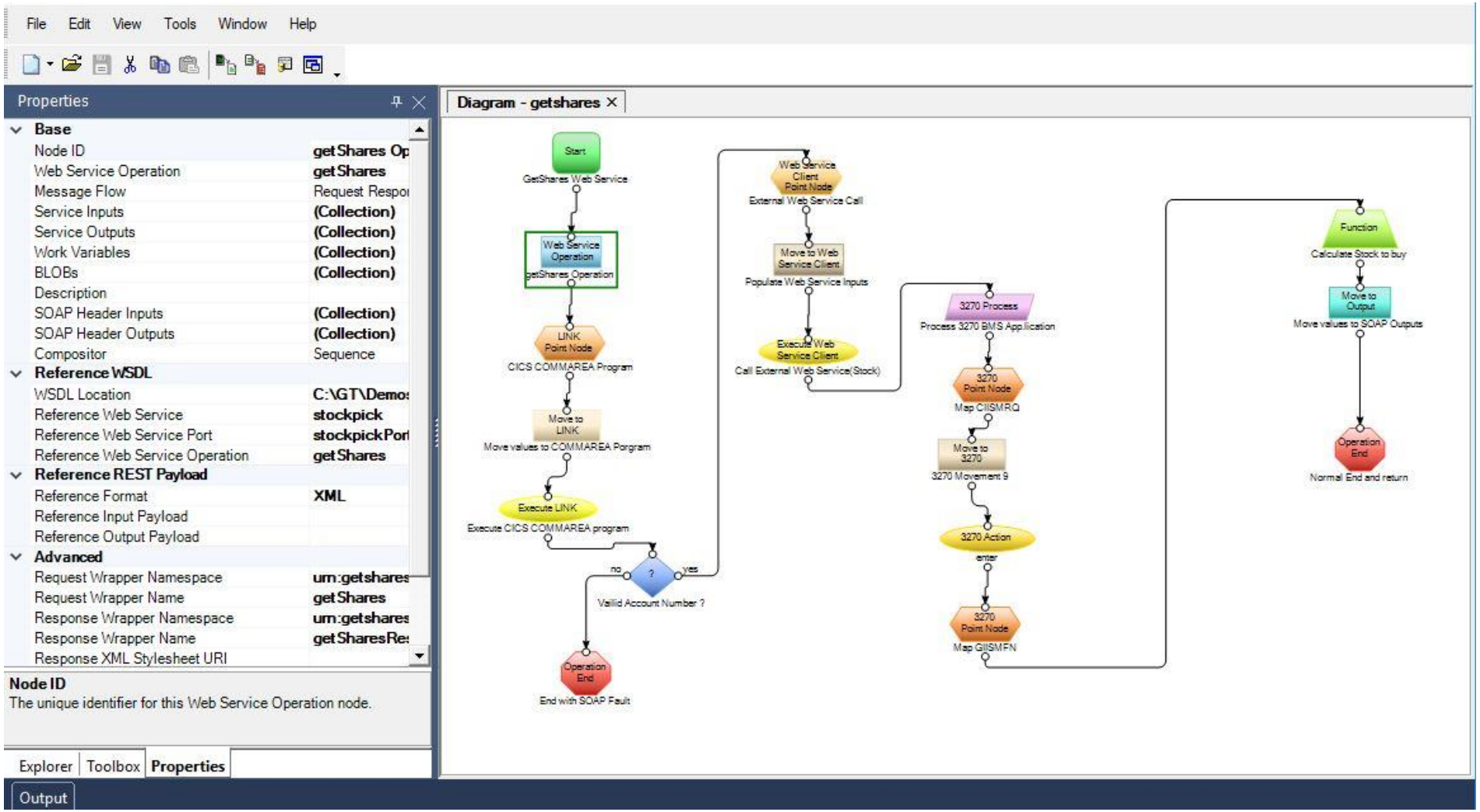
- CICS systems of record
- Instant Payment (Europe)
- Outbound calls to Google resources
- Outbound calls to Credit resources
- Outbound calls to Account Control Website
- Outbound calls to Terrorist Check sites
- Inbound API calls to existing CICS Trans with no code change
- ATM system inbound API’s (SOAP then REST)

https://maps.googleapis.com/maps/api/geocode/json?



GTSoftware®

Ivory Composite API - COMMAREA Program + Web Service + 3270 Dialog



GTSoftware®

Insurance Company

- Mid-size Property and Casualty insurer
- Headquartered in the Midwest for 115 years
- 2400+ Employees
- Processes about 10 million transactions weekly

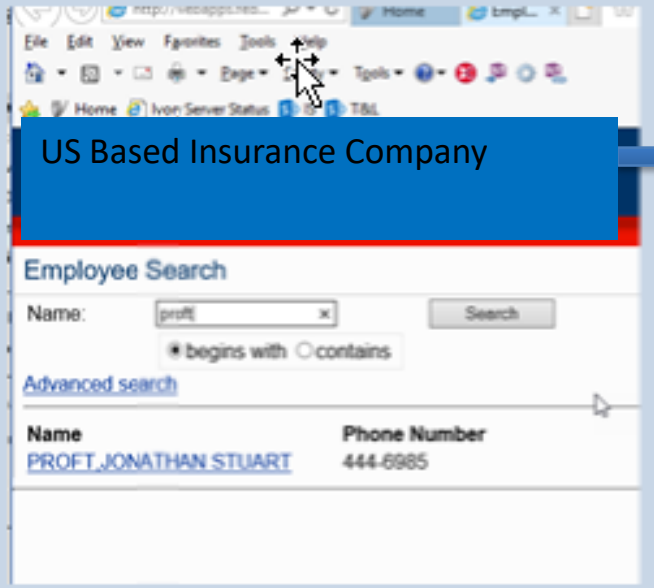
GTSoftware®

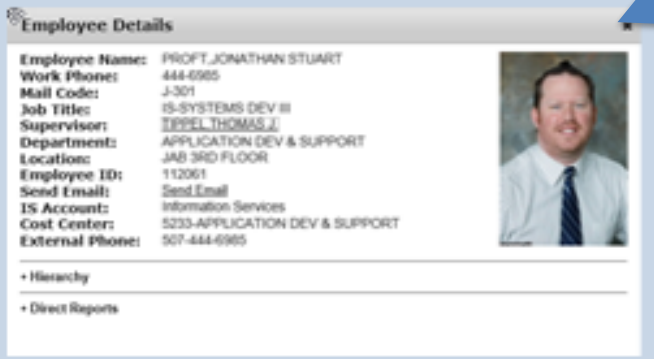
Insurance

- CICS systems of record
- Multiple Outbound calls to 3rd Party Geocode
- Calls to Legacy DB (DB2)
- CICS Links to multiple programs
- 3rd Party Lookup
- MVR Proof of Insurance
- Business Owner Policy Questionnaire
- Locate Your Rep
- Other applications
- VB Front End

GTSoftware®

Java Web App





Ivory Server

COBOL

```
LINKAGE SECTION.
01   DFHCOMMAREA.
     COPY [illegible].

PROCEDURE DIVISION USING DFHCOMMAREA.

1000-SEARCH-DATA.

     PERFORM 3000-CALL-PEOPLE-SEARCH
        THRU 3000-EXIT.

1000-PROGRAM-EXIT.
     GOBACK.

     COPY [illegible].
```

```
SELECT   [illegible]
         [illegible]
         [illegible]
FROM     [illegible]
WHERE  ((:SA-STA-CD = 'L')
         OR
        (:SA-STA-CD = 'A' AND
         ACY_IND = 'Y'))
          AND
        ((:SA-CRT1-TYP = '00')
          OR
        (:SA-CRT1-TYP = '04' AND
         ACT_CD = :SA-CRT1-VAL-AC
          OR
        (:SA-CRT1-TYP = '08' AND
         :SA-CRT1-COP = 'E' AND
         UPPER(ACT_DES) = UPPER(:
          OR
        (:SA-CRT1-TYP = '08' AND
         :SA-CRT1-COP <> 'E' AND
```

Leading Aptitude Testing Company

- U.S headquartered, non-profit assessment vendor
- Develop and administer 50 million aptitude tests annually
- 180 countries —9,000 locations

GTSoftware®

- CICS systems of record
- Multiple Outbound calls to Google Geocode
- Calls to Legacy DB (IDMS)
- CICS Links to multiple programs
- Outbound Calls for External Credit Card Processing
  - Immediate Credit Approval
  - Two Large Back-End Online Systems
  - Real-time Communication with Third-Party Credit Card Processor
- Ability to process funds for payment
- Ability to track candidate’s scheduling, testing, and scoring
- Both were green screen systems and would use same interface
- Neither coded to support encryption, SSL security and WS security tokens — a requirement for credit card processing.
  - Encrypted Security to meet PCI Compliance

GTSoftware®

GT Ivory® at One of the World's Largest Airlines

GTSoftware®

SCEPTRE

SCEPTRE stands for System Computerized for Economical Performance, Tracking, Recording, and Evaluation (airline technology)

https://www.acronymfinder.com/System-Computerized-for-Economical-Performance%2C-Tracking%2C-Recording%2C-and-Evaluation-(airline-technology)-(SCEPTRE).html

- IMS Mainframe
- Cobol
- Enterprise Solution
  - Line Maintenance
  - Base Maintenance
  - Planning
  - Finance
  - Engineering
  - Configuration Management/Records
  - Supply Chain
- Utilizing Web Front Ends

GTSoftware®

Airlines Technical Operations

- 130 IT applications supporting one of the largest airlines in the world
- Mechanics, supply chain, engineering, purchasing, planning
- Maintenance performed worldwide
- Over 10,000 Aircraft Maintenance Technicians

GTSoftware®

Environment

IMS TM & IMS DB

Ivory Service Architect (API creation & orchestration)

Github (source version control)

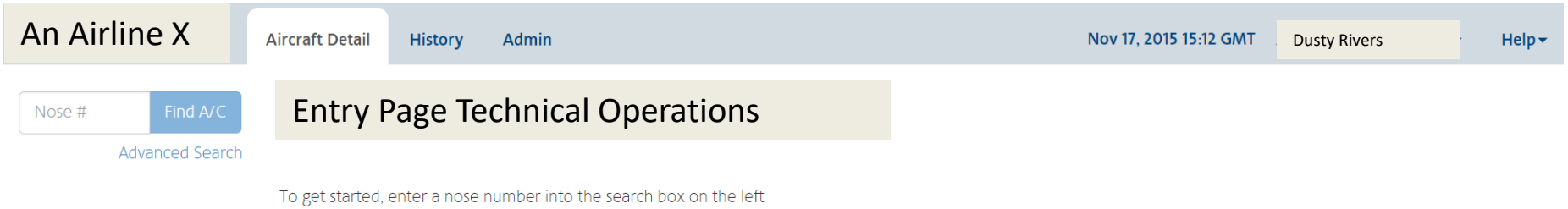
Jenkins (automation)

.NET , Java, Node.js , COBOL

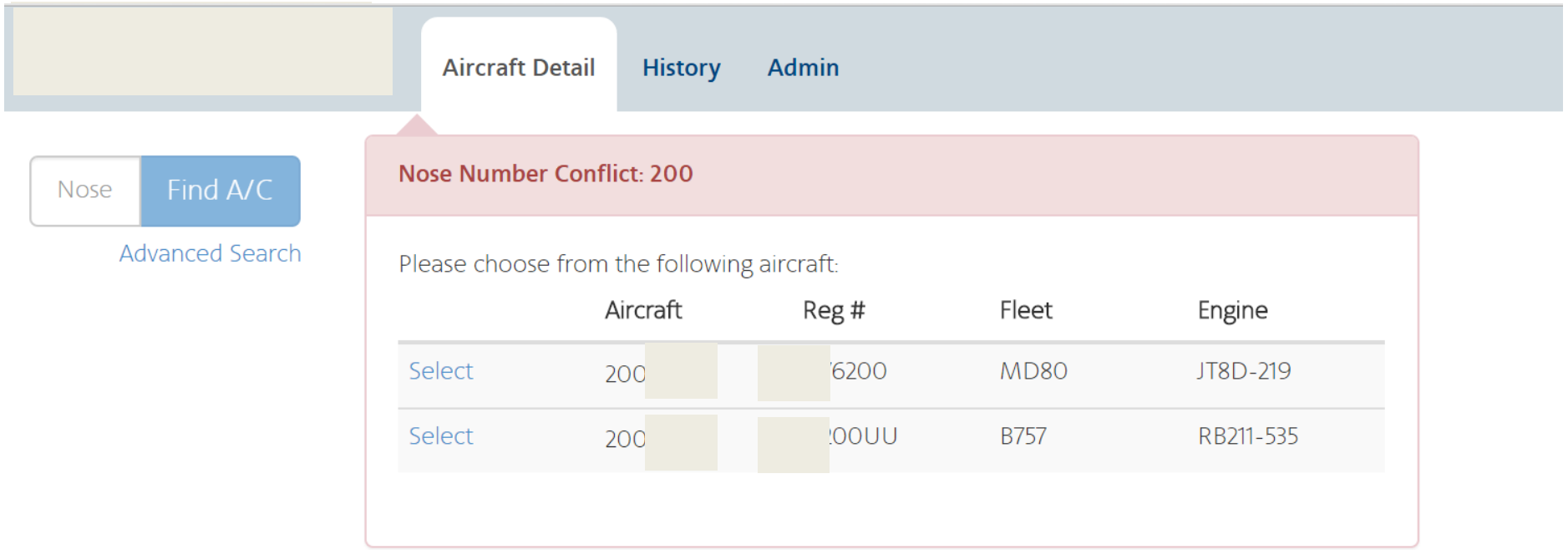
Linux (Red Hat) , JBOSS

Tomcat

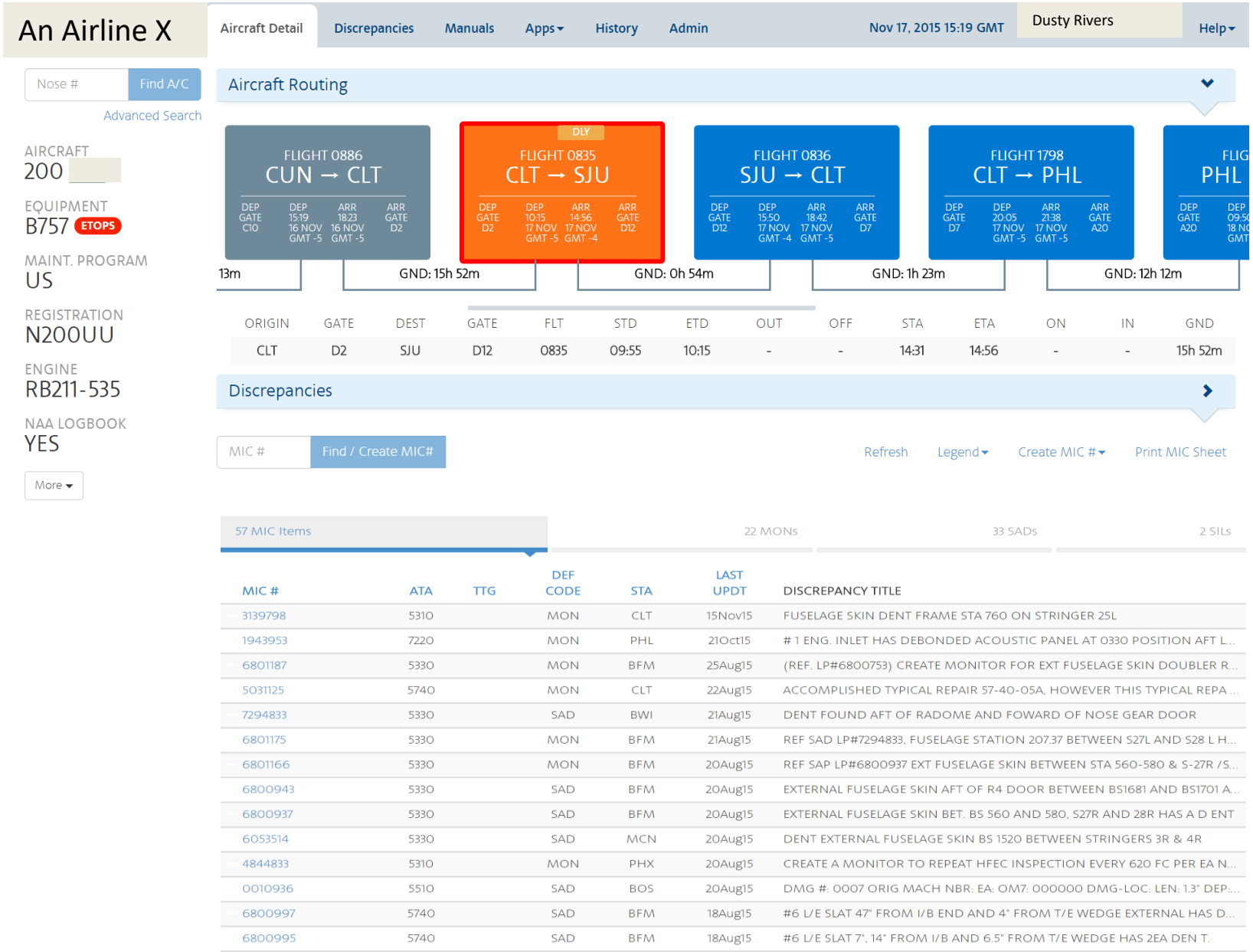
Mobile Application on Tablets & Smartphones



Nose Number Conflict Resolution



Aircraft Tail Number Selected

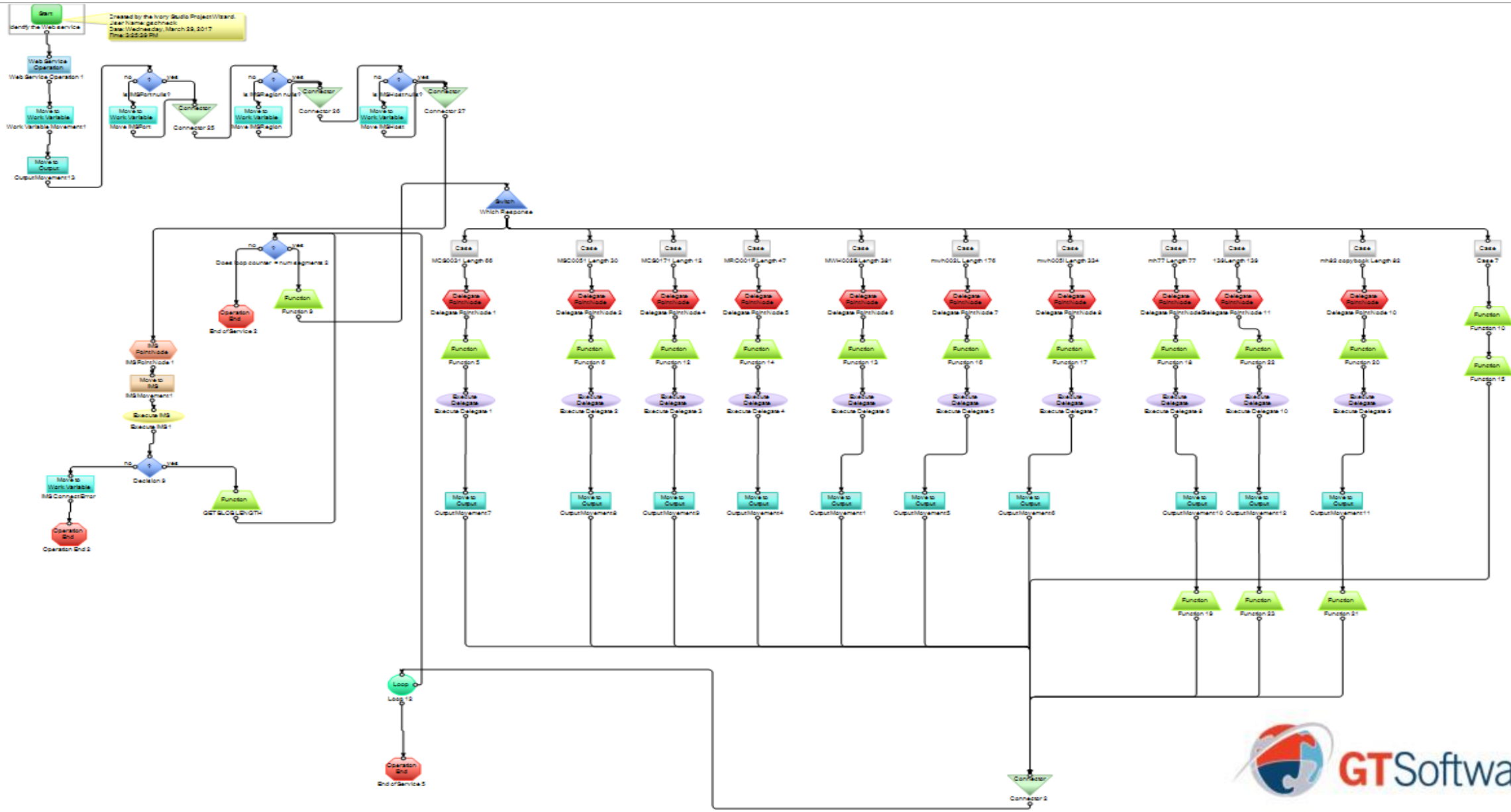


Legacy Application Complexities

- Multiple copybooks
- Multiple paths
- Conversational
- Multiple code design patterns
- Embedded screen logic

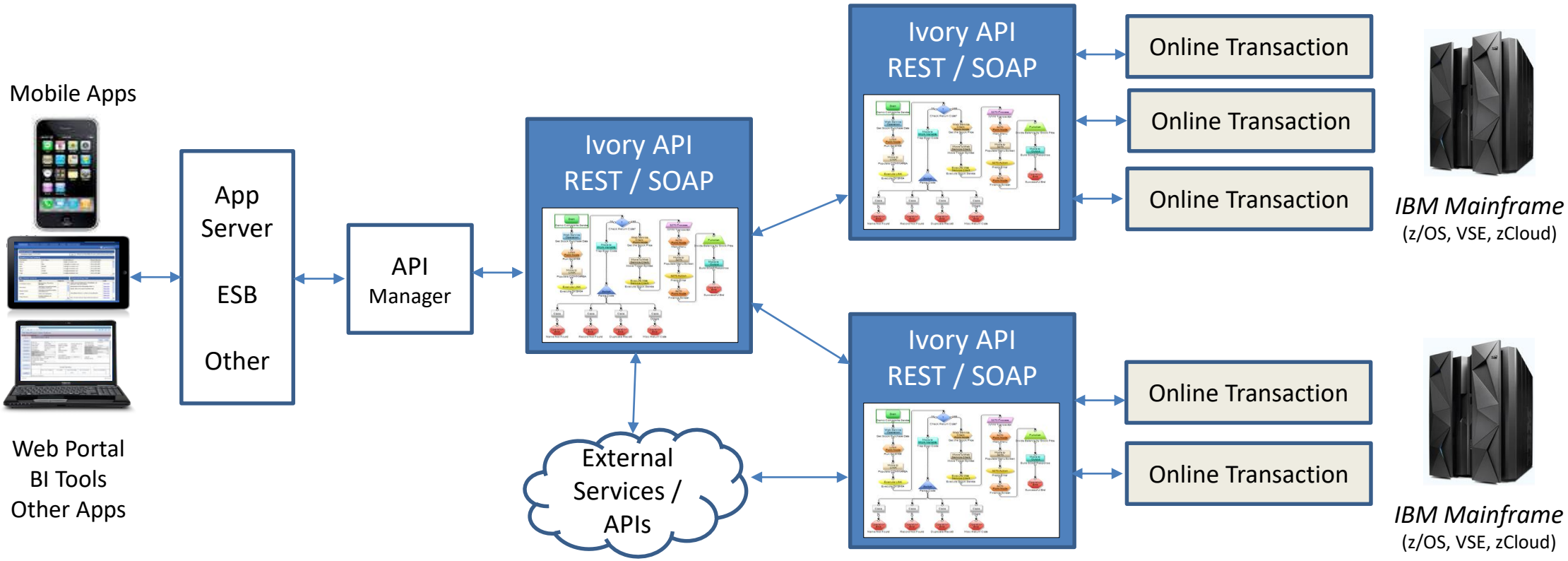
GTSoftware®

Ivory API Orchestration… No Coding!!!



GTSoftware®

GT Ivory® Mainframe APIs



GTSoftware®

GT Ivory® Deployment Options

zOS Started Task / CICS

GT Ivory Server

IFL (Linux on z)

GT Ivory Server

redhat.

SUSE

ubuntu

Windows or Linux Server

GT Ivory Server

Windows

Java

GTSoftware®

Blockchain

Business Intelligence

AI

Web

Cloud

Business Automation

Mobile

Application Integration

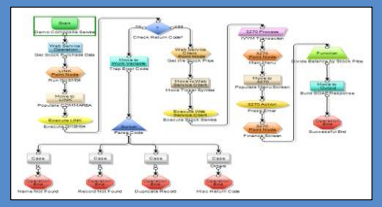
API Management

API Testing

ESB & RPA

GT Ivory®

Mainframe Transformation Gateway



API Orchestration

Mainframe Microservices and Integration Connectors

CICS Applications

IMS Applications

IDMS Applications

3270 Applications

Other Resources

Callable External Services

Zowe

DB2 REST

z/OS Connect

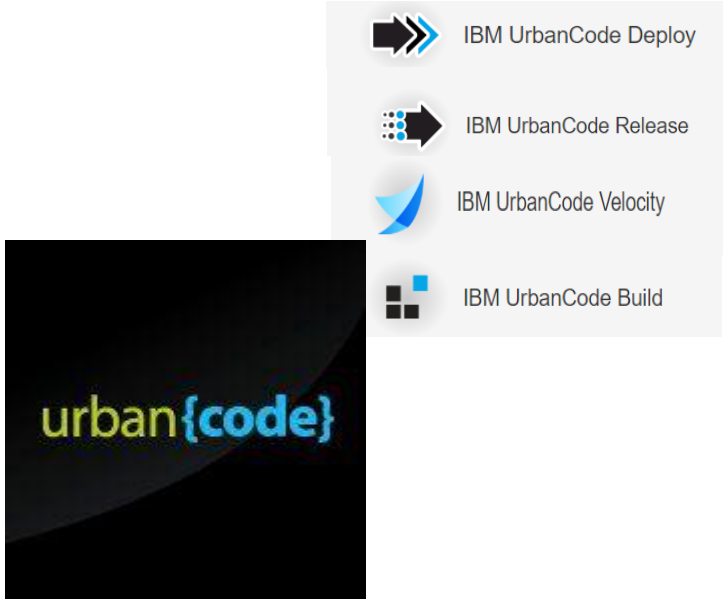
IBM DVM

GTSoftware®

What Have Customers Asked For…

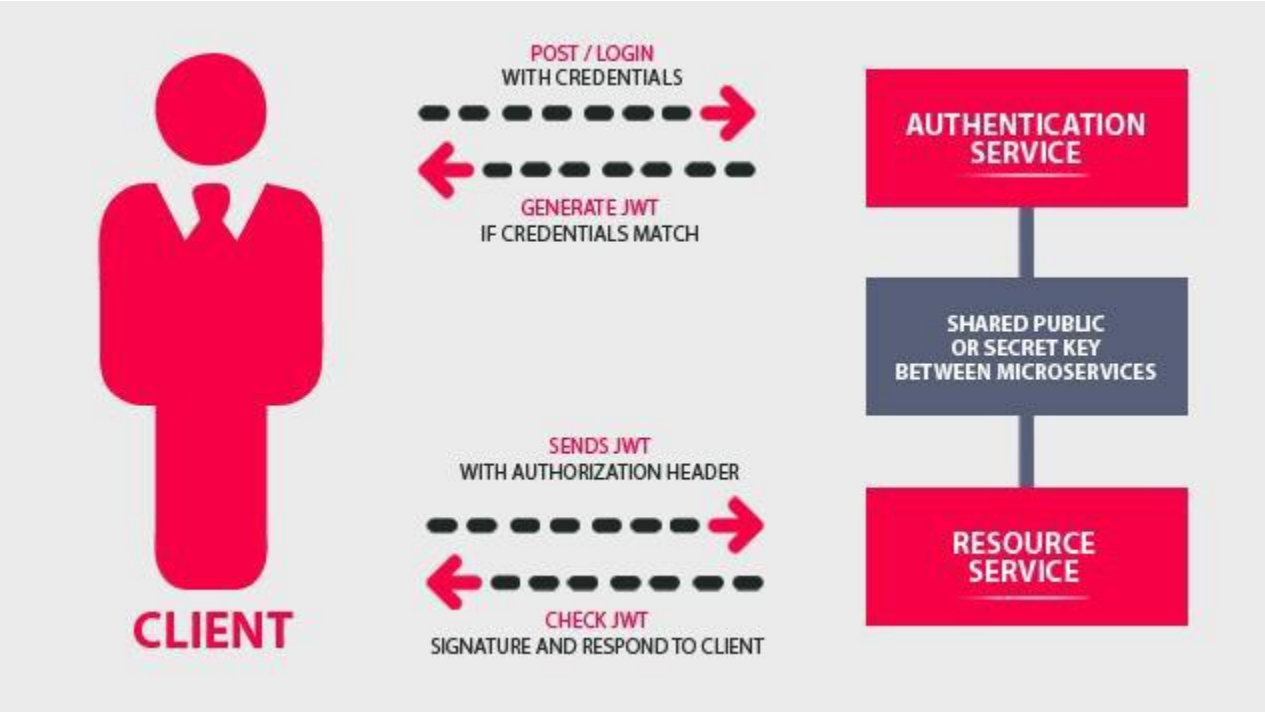
- JWT (Enhanced)
- Calling out to distributed Clients (with orchestration)
- API Repositories (which one)
- DevOps (UrbanCode
- DB2
- CLI Based Service Creation

Zowe



GTSoftware®

JWT(JSON Web Token)



GTSoftware®

JWT

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5c
CI6IkpXVCJ9.eyJzdWIiOiIxM
jM0NTY3ODkwIiwibmFtZSI6Ik
pvaG4gRG9lIiwiYWRtaW4iOnR
ydWV9.TJVA95OrM7E2cBab30R
MHrHDcEfxjoYZgeFONFh7HgQ
```

Decoded

```
{
  "alg": "HS256",
  "typ": "JWT"
}
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
)
```

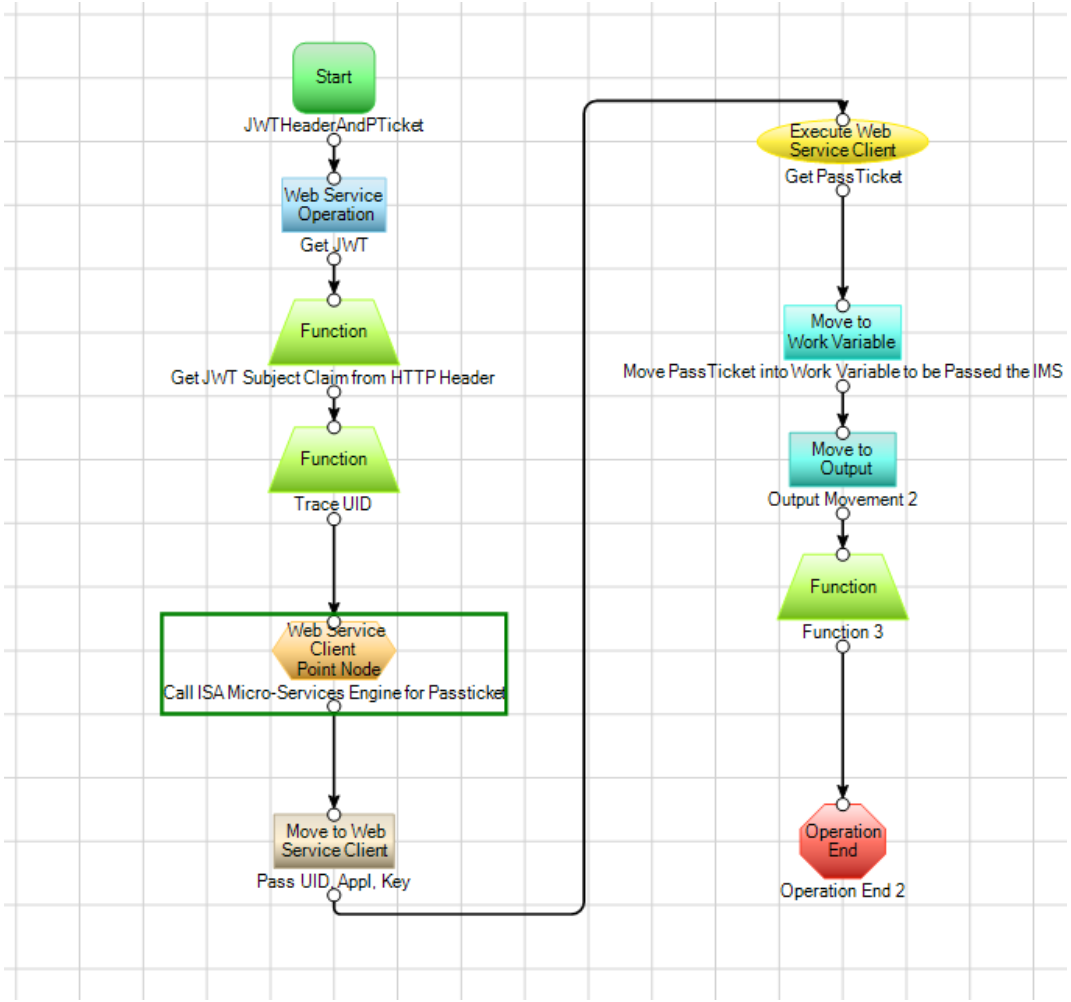
Header

Payload

Signature

GTSoftware®

JWT Sample



GTSoftware®

Callable (outbound Services)

What are Callable Services?

- Access to SOAP and JSON Services via COBOL or PL/I Call
- Call – Procedural Application Programming Interface (API)
- Used before API became a popular Web / Restful Service Term

What is needed?

- Generation of Callable Service Interface (Call) for COBOL / PL/I
- Processing of all TCPIP Services for Target Service
- Dynamic Marshaling / Parsing of all XML and/or JSON

GTSoftware®

Callable(outbound Services)

- Procedural Language API (Call)
- Procedural Language Data Layouts (Copybook)

PL/I CALL

PLI_IVORY_EXAMPLE | NameHits

```
/* ******************************************************* *
* GIISCLX module is called to set the context root.       *
* ******************************************************* */
  CONTEXT_ROOT = '/rest' || '00'X;
  CALL GIISCLX(IVORY_TOKEN,IVORY_CALL_CONTEXT,
                           CONTEXT_ROOT);
...

/* ******************************************************* *
* GIISCLX module is called to Access the Web Service      *
* ******************************************************* */
  SOAP_RETURNCODE = 0;
  CALL GIISCLX(IVORY_TOKEN,IVORY_CALL_PROCESS,
                           IVORYH,IVORY_LENGTH);
/* ******************************************************* *
* On Return the return code provides status of call       *
* ******************************************************* */
  RC = PLIRETV();
  IF RC ^= 0 THEN
     ...

  ELSE
/* ******************************************************* *
* On Good return the Web Service Data is returned...      *
* ******************************************************* */
     ...
```

REST/JSON SOAP/XML

Open Files

PLI_IVORY_EXAMPLE | NameHits | CLI.txt

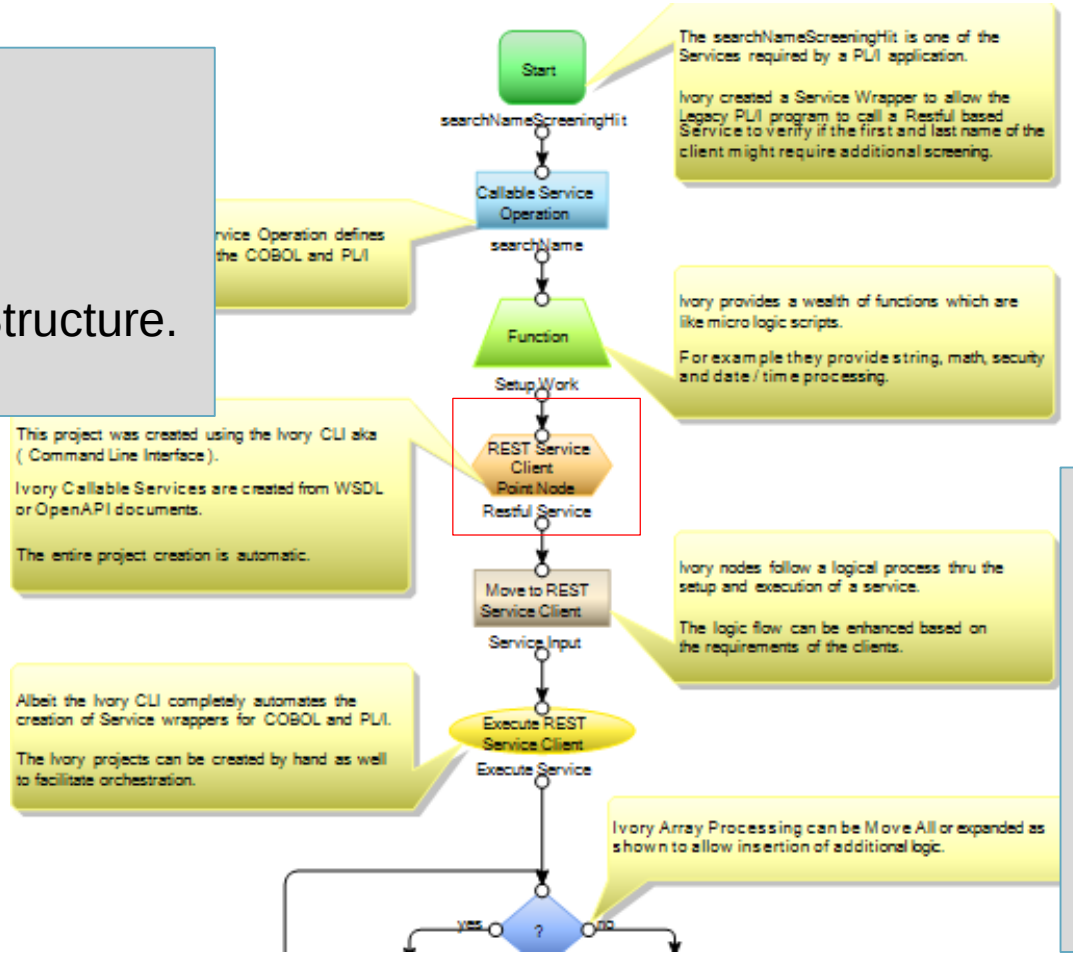
```
/* Generated by Ivory
 * Project searchNameScreeningHit
 * Date Thursday, December 6, 2018
 */
        05 SEARCHNAMESCREENINGHIT_3_0RESPONSEPAYLOAD ,
           10 ATTENTION (20) ,
              15 ATTENTIONCD CHAR(30),
              15 EXPARAMLIST (10) CHAR(30),
              15 LOGENTRY ,
                 20 LOGID CHAR(30),
                 20 LOGENTRYID CHAR(30),
           10 NAMESCREENINGHIT (20) ,
              15 CATEGORY CHAR(30),
              15 DATEOFBIRTH CHAR(17),
              15 DOMICILECD (10) CHAR(30),
              15 NAME CHAR(30),
              15 NAMEPOOL CHAR(30),
              15 NAMEPOOLCD CHAR(30),
              15 NAMERECORDREFERENCE CHAR(30),
              15 NATIONALITYCD (10) CHAR(30),
              15 PERCENTAGEOFRELEVANCE BINARY FIXED(15),
              15 RISKCATEGORY CHAR(30),
              15 RISKLEVEL BINARY FIXED(31),
              15 SUBCATEGORY CHAR(30),
        05 BUSEX ,
           10 EXCD CHAR(30),
           10 VALIDATIONERROR (1) ,
              15 VALIDATIONERRORCD CHAR(30),
              15 PARAM (1) ,
                 20 NAME CHAR(30),
```

PL/I Data Area

GTSoftware®

Callable(outbound Services)

Ivory Callable Services
access the JSON/SOAP
on processes and return
a COBOL or PL/I Data Structure.



Ivory CLI automates Callable Services creation to access external JSON/SOAP services. Additionally, the Ivory CLI will build SOAP and JSON Service wrappers for any CICS or IMS System z application.

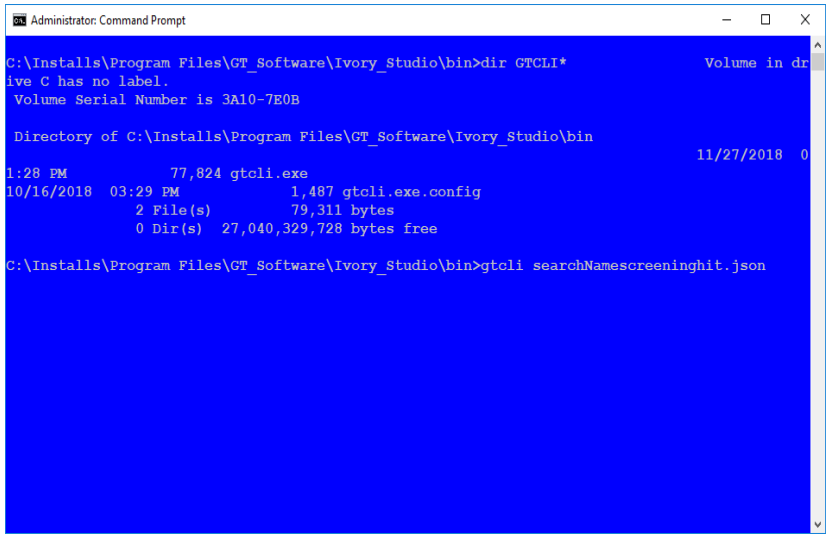
GTSoftware®

Command Line Interface (CLI)

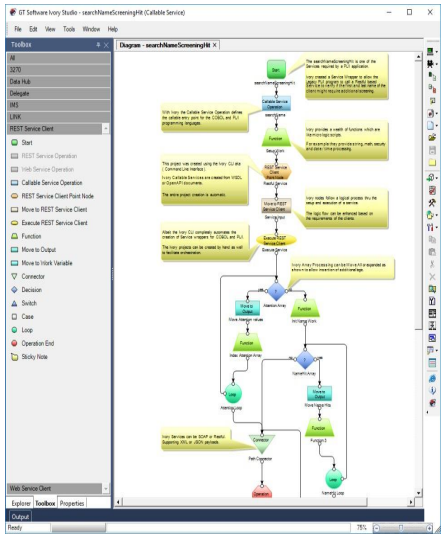
- ## Command-Line Interface

A command-line interface or command language interpreter (CLI), also known console user interface and character user interface (CUI), is a means of interacting with a computer program where the user/client issues commands to the program in the form of successive lines of text aka command lines. Commonly processed by a command language interpreter or shell interface.

CLI



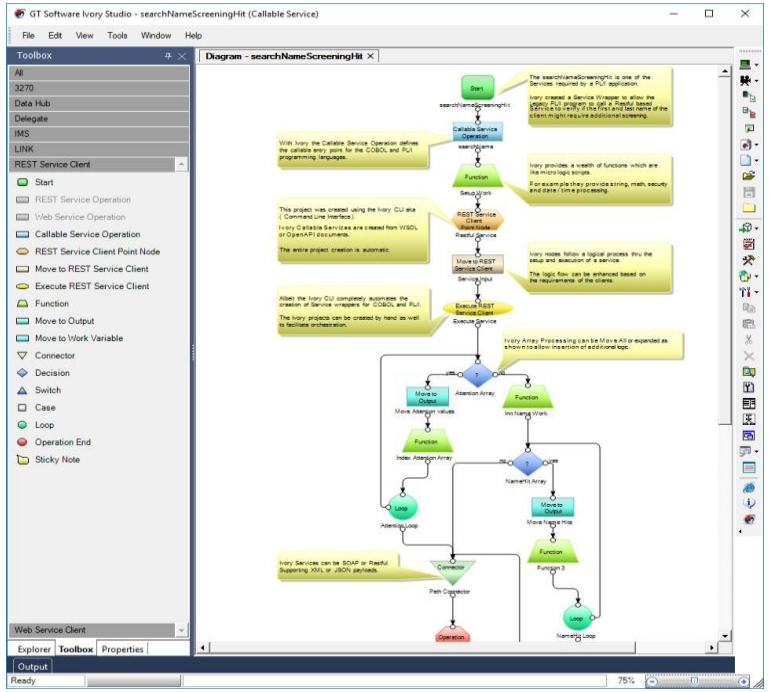
Ivory Studio



Command Line Interface (CLI)

- Command Line Interface
- Input…
  - OpenAPI (Restful JSON Services)
  - WSDL (SOAP XML Services)
- Generates Callable Services
- Removes XML/JSON Complexity
- Output…
  - Ivory Service Project

Ivory Studio



GTSoftware®

Security.....

- AT-TLS
- RACF,ACF2,Top Secret
- WS-*
- SOAP Header
- HTTP/S
- JWT (JSON Web Token)
- Passtickets

Base	
Node ID	VPSX LOGON
Service Type	SOAP
WSDL Location	file:///C:/GT%2
Encoding Style	rpc/encoded
Web Service	VPSXService
URI	https://vpsx-de
Use AT-TLS	False
Web Service Port	VPSXPort
Web Service Operation	Logon
Message Flow	Request Response
Namespace	http://www.lrs.c
Service Inputs	(Collection)
Service Outputs	(Collection)
SOAP Header Inputs	(Collection)
SOAP Header Outputs	(Collection)

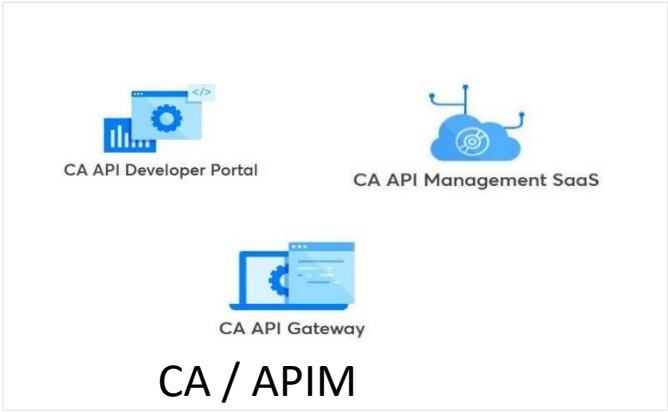
IMS Connect	
Host	@@WKHOST;
Port	@@WKPORT;
Datastore	@@WKDATASTORE;
Use Secure Connection (Java Server Only)	False
Authenticate	Use Work Variables
User ID Work Variable	WKUSERID
Password Work Variable	WKPASSWORD
User Exit	GIIIMSC2 (Default)
Commit Mode	CM1 (Send then Commit)
Synclevel	None
Timeout	IMS Connect Default
Return Code Work Variable	IMS_CONNECT_RC
Reason Code Work Variable	IMS_CONNECT_REASON_CODE
Error Text Work Variable	IMS_CONNECT_ERR_MSG
Include Each Segment LLZZ in Output Data	True
Total Length of Output Segments Work Variable	
Total Number of Output Segments Work Variable	num_segments

GTSoftware®

Where to put them……

IBM apiconnect

MuleSoft



CA / APIM

apigee

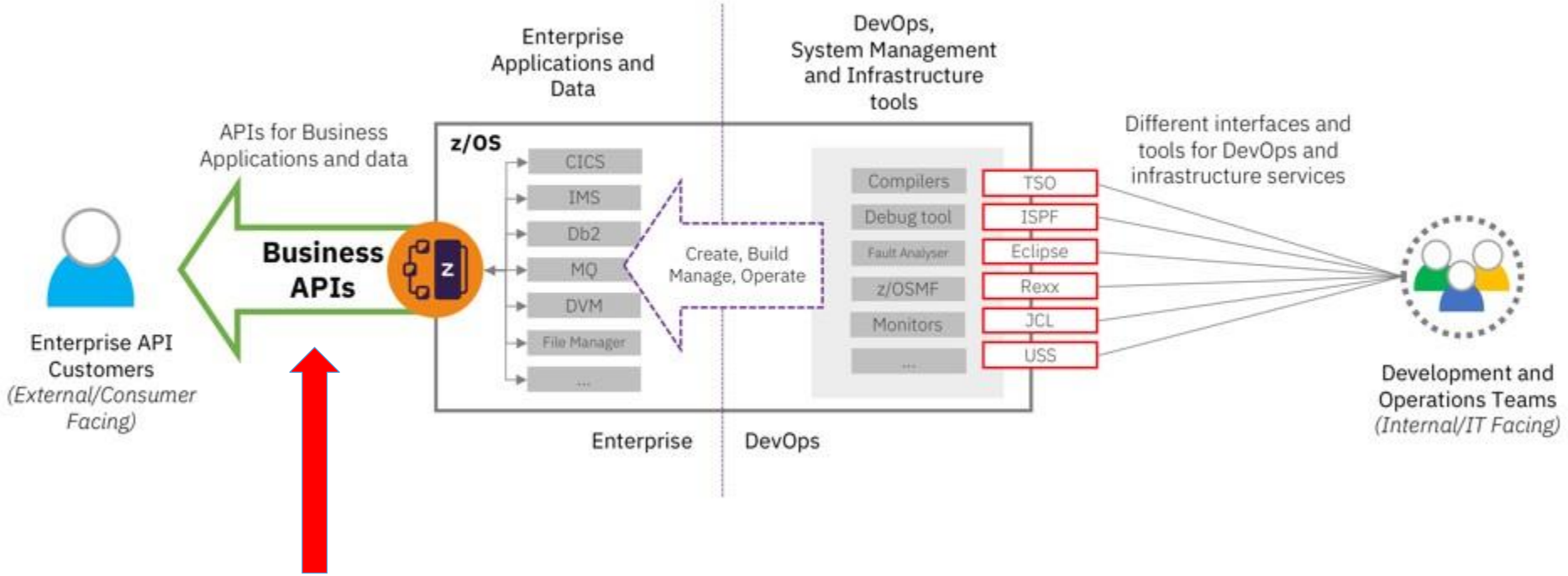
3scale BY RED HAT®

GTSoftware®

What's Next……

GTSoftware®

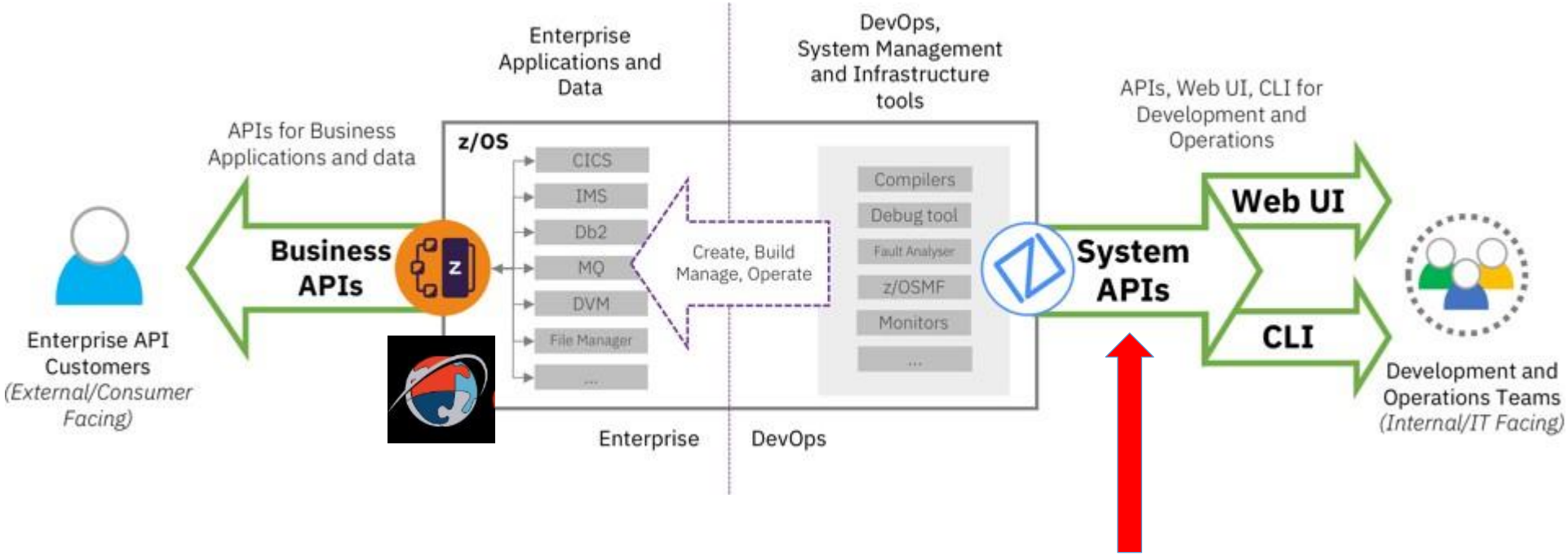
Zowe



GTSoftware®

Zowe

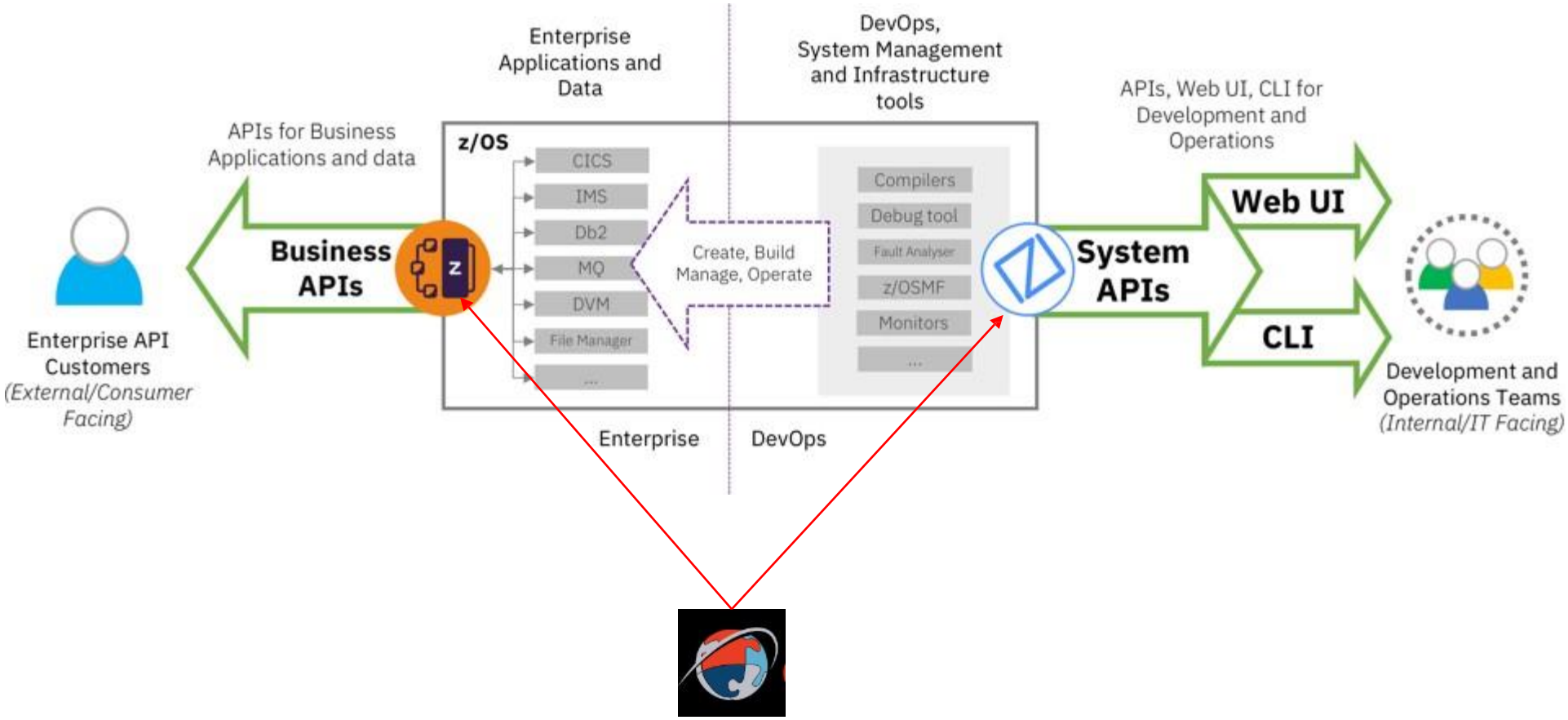
- Zowe



GTSoftware®

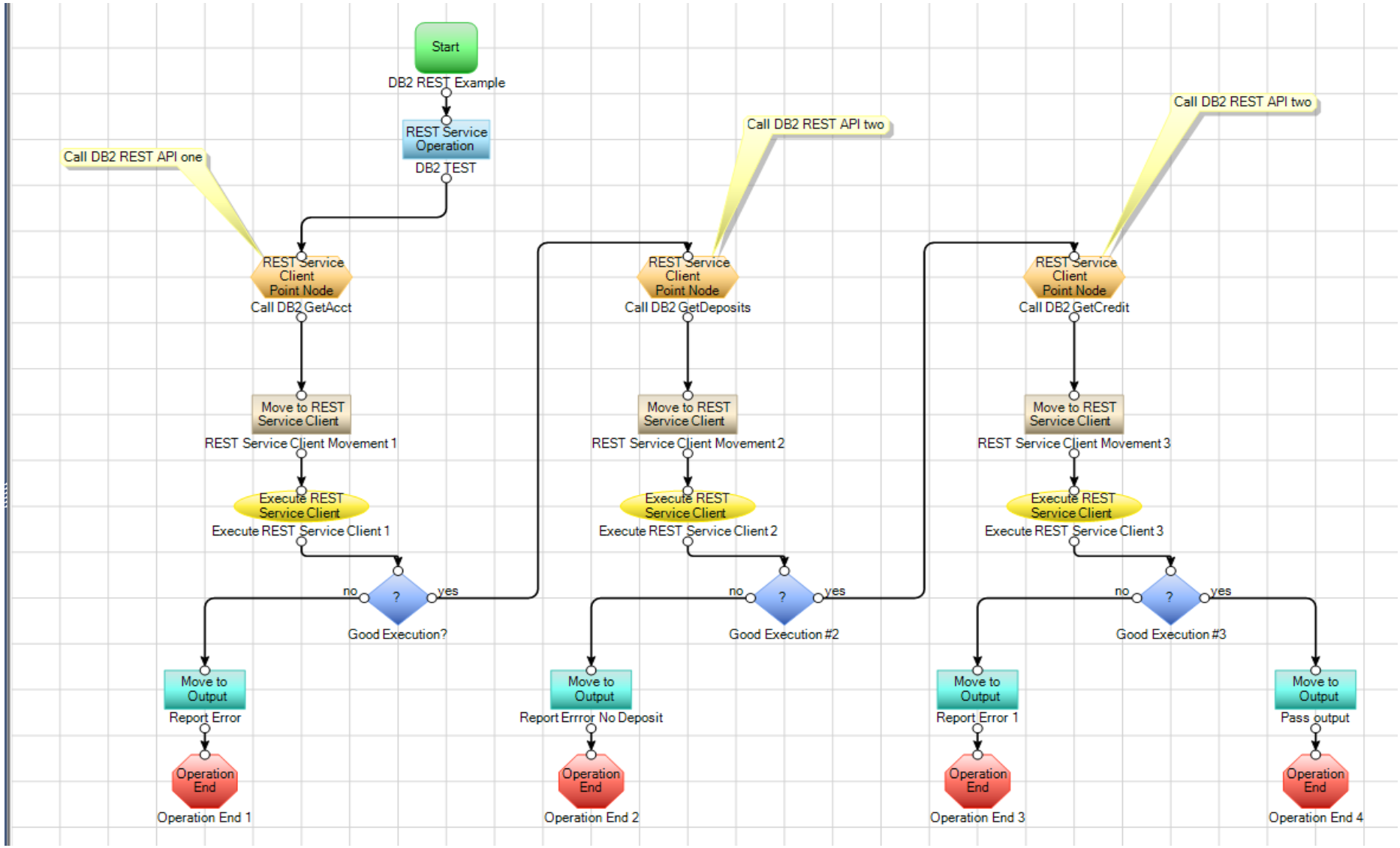
Zowe

- Zowe

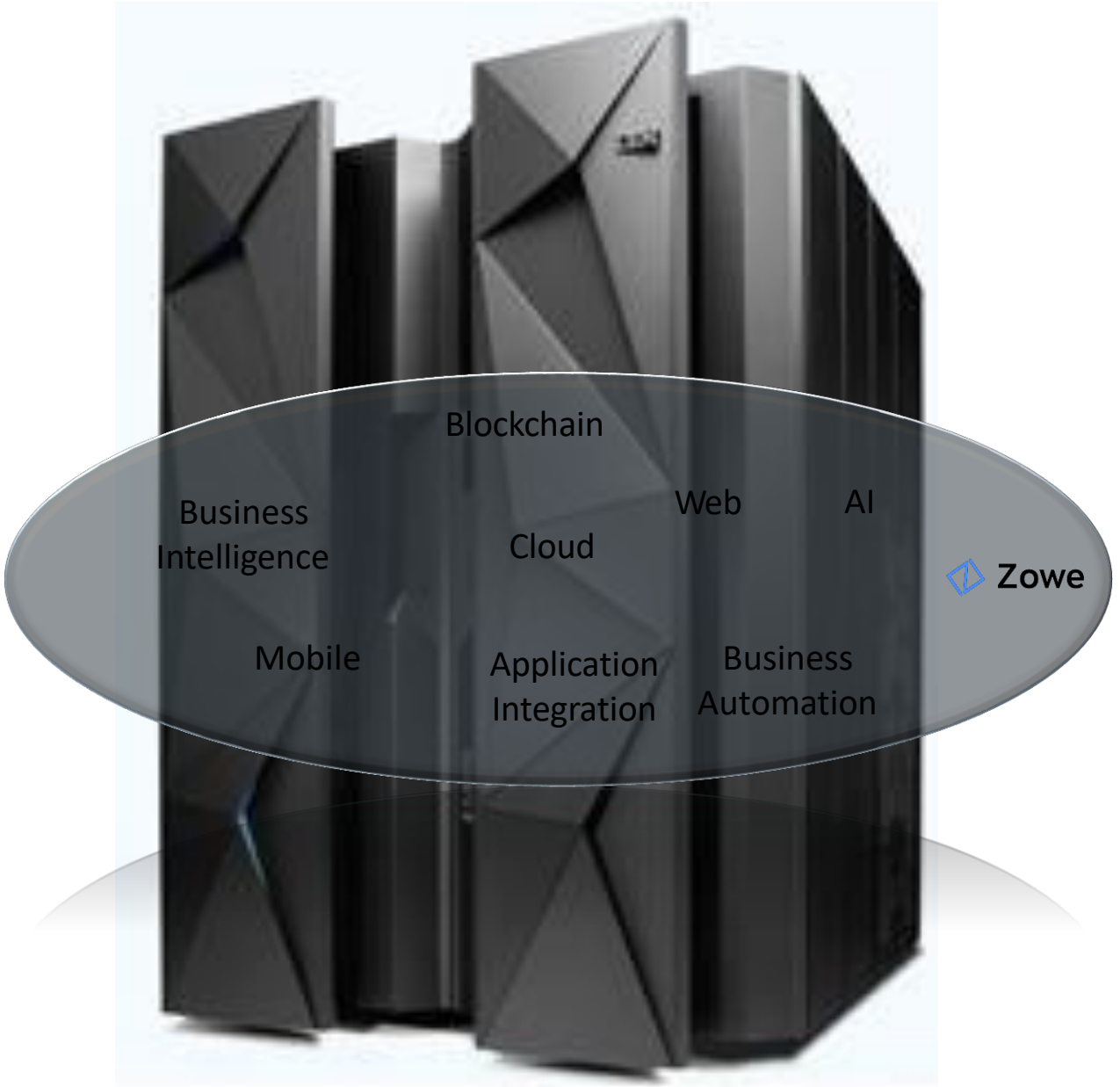


GTSoftware®

DB2 REST

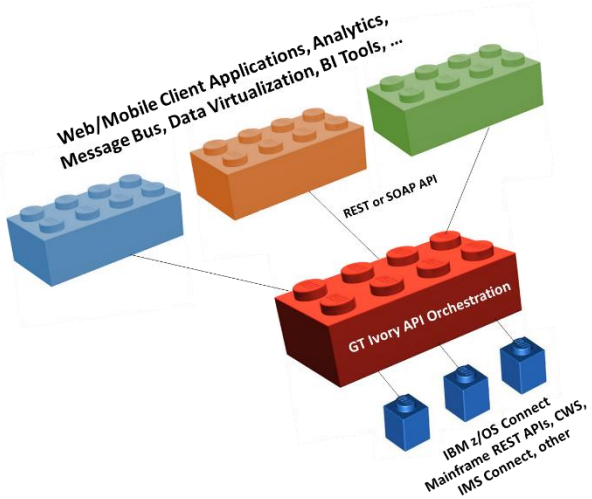

Start
DB2 REST Example
REST Service Operation
DB2 TEST
Call DB2 REST API one
REST Service Client Point Node
Call DB2 GetAcct
Move to REST Service Client
REST Service Client Movement 1
Execute REST Service Client
Execute REST Service Client 1
no
?
yes
Good Execution?
Move to Output
Report Error
Operation End
Operation End 1
Call DB2 REST API two
REST Service Client Point Node
Call DB2 GetDeposits
Move to REST Service Client
REST Service Client Movement 2
Execute REST Service Client
Execute REST Service Client 2
no
?
yes
Good Execution #2
Move to Output
Report Errror No Deposit
Operation End
Operation End 2
Call DB2 REST API two
REST Service Client Point Node
Call DB2 GetCredit
Move to REST Service Client
REST Service Client Movement 3
Execute REST Service Client
Execute REST Service Client 3
no
?
yes
Good Execution #3
Move to Output
Report Error 1
Operation End
Operation End 3
Move to Output
Pass output
Operation End
Operation End 4

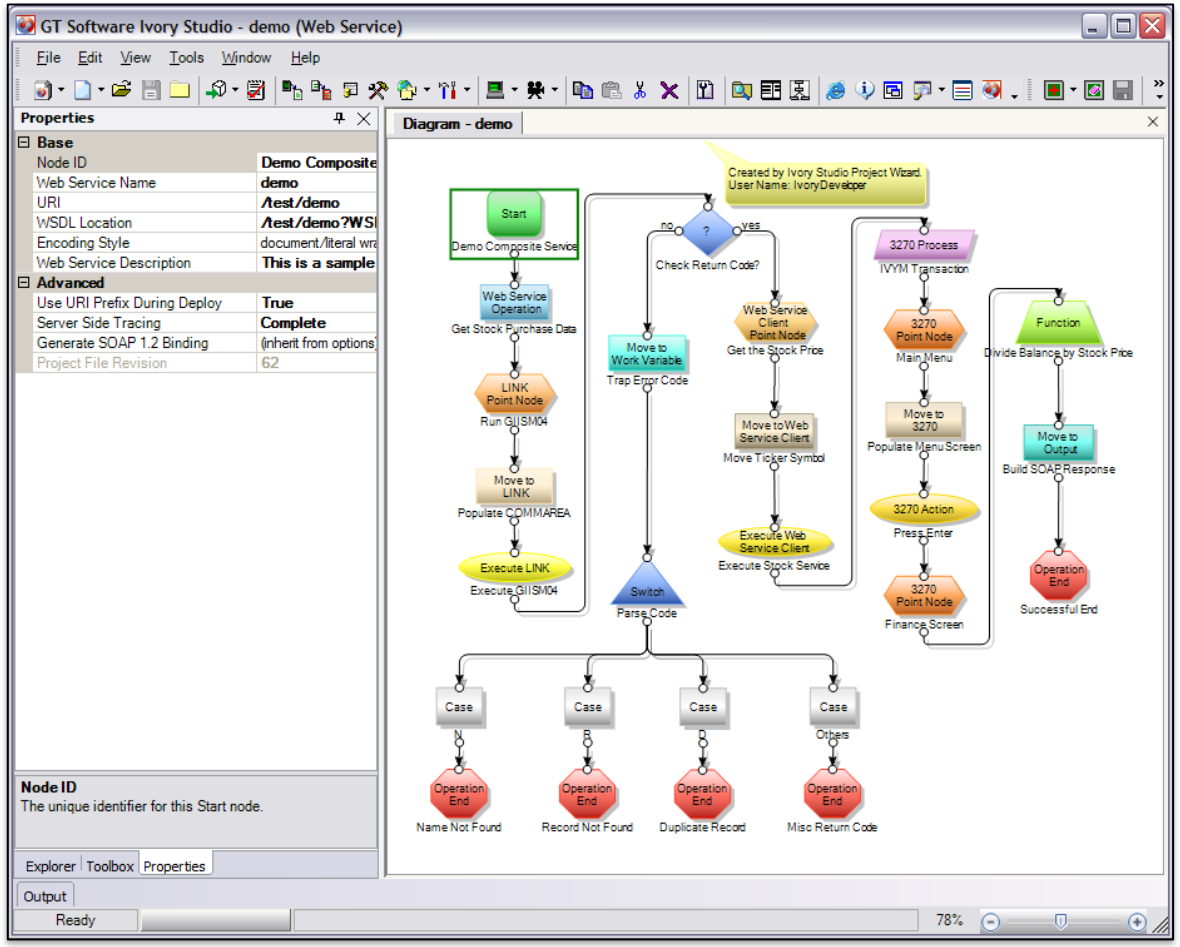
Whatever comes next……



GTSoftware®

GT Ivory Service Architect®





Intelligent Composite API:

- Multiple transactions
- Multiple data sources
- External web services and APIs
- Conditional Logic
- Error handling
- Governance and security
- Drag-and-drop (no coding) SDK
- Shared ‘business’ APIs across consumers
- No ‘low level’ coding and management of mainframe connectors
- Easy, fast, and agile development

GTSoftware®

GTSoftware®

- Founded in 1982 (HQ in Atlanta, GA)
- More than 35 years of market leadership
- Focused on real-time mainframe integration for strategic business initiatives
- Broad experience across all mainframe and distributed environments
- Worldwide cross-industry customers and strategic partnerships

www.GTSoftware.com

Lockheed Martin · AXA Technology Services (redefining / service) · Porsche · United States Postal Service · Nationwide · ETS · GE · Federated Insurance · Volvo · CareFirst · ABSA · Farm Bureau Insurance · American Airlines · ING · KPA Pension

IBM Business Partner · Microsoft Gold Certified Partner · redhat · Fujitsu Partner · Oracle PartnerNetwork · ca technologies

GTSoftware®