Introduction to CICS

Course introduction
Course introduction

- What is CICS?
- What is an application server?
- Why use an application server?
Course introduction

- Services provided by CICS
- How CICS applications are defined
- Scaling CICS to meet demand
What is CICS?

Overview of CICS
What is CICS?

- Application server on IBM Z
- Most application servers support a single language
- CICS supports many different languages
Mixed language application server

- Java
- Java Enterprise Edition
- COBOL and PL/I
- C, PHP, Assembly and others
- Third party vendor languages
Who uses CICS?

- Banking
- Transport
- Retail
- Insurance

Fortune 500 companies
Why use CICS?

Mission critical applications
- Downtime measured in $100,000+

Stock trading  Credit cards
Why do they use CICS?

Google searches:
40,000 – 60,000

> 200,000

Transactions per second
Why do they use CICS?

Scalability - is the capability of a system, network, or process to handle a growing amount of work.
Why do they use CICS?

Reliability

Reliability - an attribute of any computer-related component that consistently performs according to its specifications.
Why do they use CICS?

99.999%

Availability

Availability - the probability that a system is operational at a given time
Why do they use CICS?

- Scalability
- Reliability
- Availability

99.999%
Why do they use CICS?

- Not just for mission critical apps
- Versatility and extensibility
- See ‘How Walmart Became a Cloud Services Provider with IBM CICS’
Summary

- CICS is used in a wide range of industries
- CICS provides support for applications in a number of different languages
- CICS has exceptionally high qualities of service
What is CICS?
What is an application server?
What is an application server?

- Mixed language application server
- An application server hosts applications
Example application

Application: Book a holiday
Example application

Application: Book a holiday

Flight + Hotel
Example application

**Application:** Book a holiday

**Program:** Book Holiday

**Interface Logic**
Example application

Application: Book a holiday

Program: Book Holiday
Program: Book Flight

Interface Logic
Call Book Flight

Business Logic
Example application

Application: Book a holiday

Program: Book Holiday

Program: Book Flight

Program: Book Hotel

Call Book Hotel
Example application

Application: Book a holiday

Call Book Flight, Call Book Hotel
Return Success or Fail

Program: Book Holiday

Program: Book Flight

Program: Book Hotel
Development concerns

Transactionality

Application: Book a holiday

Program: Book Holiday

Call Book Flight,
Call Book Hotel

Return Success or Fail

Program: Book Flight

Program: Book Hotel
Development concerns

Web communication

Application: Book a holiday

Program: Book Holiday

Call Book Flight,
Call Book Hotel

Return Success or Fail

Program: Book Flight

Program: Book Hotel
Development concerns

Multiple users

Application: Book a holiday

Program: Book Holiday
Call Book Flight,
Call Book Hotel
Return Success or Fail

Program: Book Flight
Program: Book Hotel
Development concerns

Application: Book a holiday

Program: Book Holiday
Call Book Flight,
Call Book Hotel
Return Success or Fail

Program: Book Flight
Program: Book Hotel
Development concerns

Transactionality
Web communication
Multiple users
...

Application: Book a holiday

Program: Book Holiday
Call Book Flight,
Call Book Hotel
Return Success or Fail

Program: Book Flight

Program: Book Hotel
Introducing the application server

Application Server: CICS

Application: Book a holiday

Program: Book Holiday
Call Book Flight, Call Book Hotel
Return Success or Fail

Program: Book Flight
Program: Book Hotel
Introducing the API

Application Server: CICS

Application: Book a holiday

Program: Book Holiday (Java)

Program: Book Flight (C)

Program: Book Hotel (COBOL)

Call Book Flight,
Call Book Hotel

Return Success or Fail

Transactionality
Web communication
Multiple users...

API
Development concerns

Transactionality
Web communication
Multiple users
Mixed languages

Application Server: CICS

Application: Book a holiday

Call Book Flight, Call Book Hotel
Return Success or Fail

Program:
Book Holiday (Java)

Program:
Book Flight (C)

Program:
Book Hotel (COBOL)
Summary

- Other concerns beside business logic
- Application server handles these other concerns on behalf of the application
- Faster time to value
- Provided for mixed language applications
CICS as an application server

Section overview
CICS as an application server

- Connection technologies
- Security
- Passing data between programs
- Accessing databases
- Monitoring
- Transactions
Summary

- Services CICS provides to applications
- CICS uniquely provides these to mixed language applications
CICS as an application server
Connecting to your application
Connecting over the web

Application: Book a holiday

Application Server: CICS

Application: Book a holiday

Book Holiday
Java program

Book Flight
C program

Book Hotel
COBOL program
Connecting via a queue

Application: Book a holiday

Book Holiday
Java program

Book Flight
C program

Book Hotel
COBOL program
Connecting via a queue

Application Server: CICS

Application: Book a holiday

- Book Holiday
  - Java program

- Book Flight
  - C program

- Book Hotel
  - COBOL program
Different data formats

Application: Book a holiday

JSON

Application Server: CICS

Application: Book a holiday

Book Holiday
Java program

Book Flight
C program

Book Hotel
COBOL program

{
  "destination": "New York"
}
Different data formats

Application: Book a holiday
XML
Application Server: CICS
Book Holiday
Java program
Book Flight
C program
Book Hotel
COBOL program
<destination>
  New York
</destination>
CICS data transformation

Application: Book a holiday

Application Server: CICS

Application: Book a holiday

- Book Holiday (Java program)
- Book Flight (C program)
- Book Hotel (COBOL program)

Input Formats:
- JSON
- XML

Output Formats:
- JSON
- XML
CICS data transformation

JSON
{
  "destination": "New York"
}

XML
<destination>
  New York
</destination>

Application Server: CICS

Application: Book a holiday

- Java program
- C program
- COBOL program

Native data

Book Holiday
Book Flight
Book Hotel
CICS data transformation

- Faster time to value
- No code changes, less risk
- Connect existing programs to new technologies
Summary

- CICS provides a range of connection technologies
- Data transform technologies
- Efficient reliable connection
CICS as an application server

Securing your application
Securing your application

- Identification and authentication
- Authorization
- Encryption
Identification and authentication

User: Alice
Password: passw0rd

User: Bob
Password: letmein

Application Server: CICS

Application: Book a holiday

- Book Holiday
  - Java program

- Book Flight
  - C program

- Book Hotel
  - COBOL program
Identification and authentication

- Username and password
- SSL certificate
- Open Authorization
- CICS handles authenticating the token
Identification and authentication

- Is the identification provided ‘valid’
- CICS shares the System Authentication Facility of z/OS
- Single security control point for the entire platform
Can this user access this resource

Can this user perform this action

Authorization rules held in SAF

Applied to individual users or groups of users
Encryption

User: Alice
Password: passw0rd

User: Bob
Password: letmein

Application Server: CICS

Application: Book a holiday

- Book Holiday
  Java program

- Book Flight
  C program

- Book Hotel
  COBOL program
Summary

- Identification and authentication
- Authorization
- Encryption
- Security single point of control
CICS as an application server
Calling programs and transferring data
Calling a program

Application Server: CICS

Application: Book a holiday

- Book Holiday: Java program
- Book Flight: C program
- Book Hotel: COBOL program
Calling a program in the same language

Application Server: CICS

Application: Book a holiday

- Book Holiday
  - Java program

- Book Flight
  - Java program

- Book Hotel
  - Java program
Calling a program as an external service

Application Server
Book Holiday
Java program

Application Server
Book Flight
C program

Application Server
Book Hotel
COBOL program

XML

JSON
Calling a program as an external service

- Complex maintenance and administration
- Inefficient solution
CICS API

- Language independent
- Allows one program to call another
- CICS LINK API
- CICS handles all the mixed language concerns in a robust manner
CICS link with data

From: LHR
To: LDW
Date: 14/01/17
Time: 1300

Application Server: CICS

Application: Book a holiday

Book Flight
C program

Book Holiday
Java program

Near: LDW
Date: 14/01/17
Nights: 5
Stars: 3

Book Hotel
COBOL program
Channel and containers

- Channels and containers used to pass data between programs
- Container – named data area
- Channel – a collection of containers
- The CICS LINK API call provides a parameter to specify a channel name
# Calling a program and passing data

## Flight-channel

<table>
<thead>
<tr>
<th>From-Airport</th>
<th>Dest-Airport</th>
<th>Date</th>
<th>Time</th>
<th>Booking-ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHR</td>
<td>LDW</td>
<td>14/01/17</td>
<td>1300</td>
<td>ABCDE1234</td>
</tr>
</tbody>
</table>

- **Book Holiday Java program**
- **Book Flight C program**
Summary

- Programs can call others irrespective of language
- Mixed language application server
- New programs can easily re-use existing business logic
CICS as an application server

Accessing databases
Accessing databases

Application Server: CICS

Application: Book a holiday

- Book Holiday
  Java program

- Book Flight
  COBOL program

- Book Holiday
  Java program
CICS is not a database

- Efficient connectors to databases
- Use language specific APIs for access
When things go wrong

Application Server: CICS

Application: Book a holiday

- Book Holiday: Java program
- Book Flight: C program
- Book Hotel: COBOL program
Summary
CICS as an application server

Monitoring
Monitoring an application

- Service Level Agreements
- Application use
- Capacity planning
Monitoring use

Application: Book a holiday
Application Server: CICS

Application: Book a holiday
Book Holiday
Java program

Book Flight
C program

Book Holiday
COBOL program
Monitoring use

- Language independent monitoring
- Monitoring records written to a central location
Monitoring use

Time of day

Number of users
Monitoring use

Application: Book a holiday

Application Server: CICS

Application: Book a holiday
- Book Holiday Java program
- Book Flight C program
- Book Holiday COBOL program
Pushing to the limit

Application: Book a holiday
Monitoring Records
Monitoring Reports
Book Holiday
Java program
Application Server: CICS
Book Flight
C program
Database
Book Holiday
COBOL program
Summary

- CICS provides functions to measure application performance
- Measure performance over a mixed language application
- No changes to program source code
- Monitor the whole application, not just the sum of its parts
CICS as an application server

Transactions
Handling problems

Application Server: CICS

Application: Book a holiday

- Book Flight: C program (correct)
- Book Holiday: Java program
- Book Hotel: COBOL program (incorrect)
Handling problems

Application Server: CICS

Application: Book a holiday

Unit of work

Book Holiday
Java program

Book Flight
C program

Book Hotel
COBOL program
Extended transactions

- Transactionality applies to all resources
- All requests processed within a unit of work
- All updates are in doubt until the transaction is completed
Syncpoint API

- Syncpoint
- Rollback
Mixed language transactions

- Transactions are not unique to CICS
- Java Transaction Architecture (JTA)
- Maintain transaction over mixed language applications
Summary

- Transactions are important
- Each request is a transaction
- Implicit transaction model
- Transactions supported over mixed languages
CICS as an application server

Summary
Summary

- CICS provides services to applications
- Single or mixed languages applications
- Extend existing applications
- Operate application as a single entity
Configuring an application in CICS

Introduction to resource definitions
Example application

- User provides holiday information
- Holiday is booked
- User receives response
Example application

Application: Book a holiday
Application Server: CICS
Program: Book Flight
Program: Book Hotel
TCP/IP SERVICE
URIMAP
PROGRAM
Program: Book Holiday

Program: Book Flight
Program: Book Hotel
Handling a request from the web

Application: Book a holiday
Application Server: CICS
Program: Book Flight
Program: Book Hotel
Program: Book Holiday (Java)
Example web request

http://cics.ibm.com:80/Holiday/AliceInfo

Hostname = http://cics.ibm.com

Port = 80

Request = Holiday/AliceInfo
Listening for a web request

Resource Definition: TCPIPSERVICE

- Defines the port for CICS to listen on
- Can specify security options such as to use SSL encryption

Application Server: CICS
Reaching our application

Resource Definition: URIMAP

- Routes a request to target program
  
  http://cics.ibm.com:80/Holiday/AliceInfo

- ‘/Holiday/*’ to Book Holiday

Application Server: CICS

Program:
Book Holiday (Java)
Our application program

Resource Definition: PROGRAM

- Reference to the actual program
- Describes how and where the program should execute

Application Server: CICS
Program: Book Holiday (Java)
What has happened so far

- User has sent a request to CICS over the web
- CICS has listened for and routed that request to our program
What happens next

- Our program will process the request and respond to the user using the same communication channels and protocols.

Application Server: CICS
Program: Book Holiday (Java)
Resource definitions

- Layer of abstraction above underlying resources
- Application: program
- Infrastructure: HTTP connection
Why use resource definitions?

- Allow configuration of various application and system components
- Programs refer to resource definition not to the underlying resource
  - Underlying resource can be changed without needing to alter the program
Example

- Replace ‘Book Hotel’ with Java program
- ‘Book Holiday’ can remain unaltered
- Calls Book Hotel through the CICS API

Application: Book a holiday

- Book Holiday
  Java program
- Book Flight
  C program
- Book Hotel
  COBOL program
Summary

- TCPIPSERVICE
- URIMAP
- PROGRAM
- Other definitions available to configure other capability
Configuring an application in CICS

Working with resource definitions
Resource definitions

- CICS provides interfaces to create resource definitions
- Definitions can be grouped together
Example application
Example application

TCPIPSERVICE

URIMAP

PROGRAM

PROGRAM

PROGRAM

DATABASE
Resource definitions

- Installing resources, installs the application
- Installed resources control the application
PROGRAM resource

- USE COUNT – how many times the program has been used
- NEWCOPY – reload the program into CICS
TCPIPSERVICE resource

- Temporarily disable new requests
- Change the security profile
Resource definitions are used to install applications

Resource definitions can also control applications
Scaling a CICS application

Introduction to scaling
Introduction to scaling
Example application

Application Server: CICS

Application: Book a holiday

Program: Book Flight

Program: Book Hotel
Example application

CICS region

**Application:** Book a holiday

**Program:**
- Book Holiday
- Book Flight
- Book Hotel
Example application

CICS region

**Application:** Book a holiday

- **Program:** Book Flight
- **Program:** Book Hotel

**Application:** Book a cruise

- **Program:** Book Day Trip
- **Program:** Book Holiday
- **Program:** Book Cruise
Enough is never enough

- Running a region at maximum is not always optimum
- Single point of failure
- CICS applications can span multiple regions
Example application

CICS region

Application: Book a holiday

Program: Book Holiday

CICS region

Application: Book a cruise

Program: Book Day Trip
Program: Book Hotel
Configuration changes

▶ REMOTESYSTEM attribute
▶ No program source code changes required
▶ Channel and containers are shipped to the remote region
Summary

- CICS can scale to run very large workloads
- Programs can be run across multiple CICS regions
- How to best split an application over multiple CICS regions
Scaling a CICS application

Splitting an application into layers
Applications split into layers

- Each layer provides a different capability
- Interface logic
- Business logic
- Each layer is a subset of programs
Interface logic

Interface Logic

Business Logic
Interface logic

Business Logic

Web
Mobile
Business Partner
Business logic

Web

Web Service

REST API

Business Logic
Naming conventions

WOR

WOR

WOR

AOR
Why split into layers

- Different layers require different runtime characteristics
- Each request might spend more time in certain layers
- Each layer can be scaled independently
Splitting the example into layers

**CICS region**

**Application:** Book a holiday

- **Program:** Book Flight
- **Program:** Book Hotel
- **Program:** Book Holiday

- **Interface Logic**
- **Business Logic**
Book a holiday – in two layers

Program:
Book Flight

Program:
Book Hotel

Business Logic

Interface Logic

CICS region

Program:
Book Holiday

CICS region

Program:
Book Flight

Program:
Book Hotel
Example application

CICS region

Program: Book Holiday

CICS region

Program: Book Flight
Program: Book Hotel

CICS region

Program: Book Flight
Program: Book Hotel
Example application
Example application

- **CMAS**
  - Program: Book Flight
  - Program: Book Hotel

- **CICS region**
  - Program: Book Holiday
  - Program: Book Flight
  - Program: Book Hotel

- **CICS region**
  - Program: Book Flight
  - Program: Book Hotel
Summary

- Applications split into layers
- Each layer runs in one or more region
- CMAS can route work to any of the regions
Scaling a CICS application

Scaling for resilience
Scaling for resilience

Program: Book Flight
Program: Book Hotel

CICS region

Program: Book Holiday

CICS region

Program: Book Flight
Program: Book Hotel
Scaling for resilience

Program: Book Flight

Program: Book Hotel

Program: Book Holiday

CMAS

CICS region
Scaling for resilience

Program: Book Flight
Program: Book Hotel

CICS region

Program: Book Holiday

CMAS
Scaling for resilience

CICS region

Program: Book Flight

Program: Book Hotel

CICS region

Program: Book Holiday

TCPIPSERVICE

CICS region

Program: Book Holiday

TCPIPSERVICE

CICS region

Program: Book Flight

Program: Book Hotel

CICS region

Program: Book Flight

Program: Book Hotel
Scaling for resilience

- LPAR – single instance of z/OS
- A single mainframe can support multiple LPARs
- CICS can be distributed across LPARs
Scaling for resilience
Scaling for resilience

LPAR 1

CICS region
CICS region
CICS region
CICS region

LPAR 2

CICS region
CICS region
CICS region
CICS region
Scaling for resilience
Single system image

Define/install/administer resource

CMAS

CICS region

CICS region

CICS region
Summary

- CICS applications scale across servers, operating systems and hardware
- Workload can be maintained during upgrading or system failure
- 99.999% reliability
Summary
Course review
Course review

- CICS is an application server on z Systems
- Used by majority of Fortune 500 companies
- Runs mission critical applications
Course review

- CICS provides all the capabilities of an application server
- CICS can host applications consisting of programs written in different programming languages
- Single API allows programs to call and share data
Course review

- As your business scales so does your application with CICS
- CICS can provide resilience across many instances
- Single system image allows you to control all instances from a single interface
Course review

- CICS is an unparalleled mixed language application server
- Security, transactionality, management, monitoring are independent of programming language