Introduction to CICS

Pradeep Gohil
IBM

November 2018
Session AI

Please note…

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

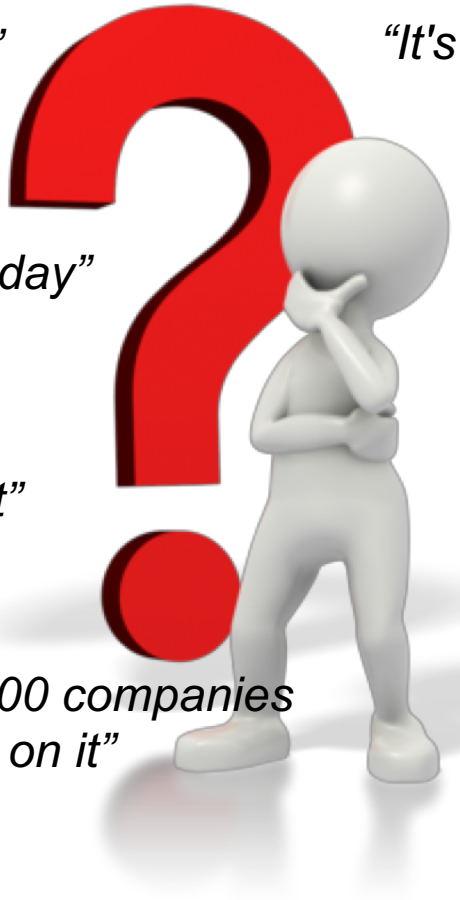
The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

CICS is...

"It's really old"

"It's very reliable and secure"



"You use it every day"

"It makes lots of money"

"It's really fast"

"It does banking"

"Most Fortune 500 companies depend on it"

"It runs the world"

For a better answer we turn to Wikipedia...

"Customer Information Control System (CICS) is a transaction server that runs primarily on IBM mainframe systems under z/OS and z/VSE.

"CICS is a transaction manager designed for rapid, high-volume online processing."

"CICS manages the entire transaction such that if for any reason a part of the transaction fails all recoverable changes can be backed out."

For a better answer we turn to Wikipedia...

*"Customer Information Control System (CICS) is a **transaction** server that runs primarily on IBM mainframe systems under z/OS and z/VSE.*

*"CICS is a **transaction** manager designed for rapid, high-volume online processing."*

*"CICS manages the entire **transaction** such that if for any reason a part of the transaction fails all recoverable changes can be backed out."*

So what is transaction processing?

"In computer science, transaction processing is information processing that is divided into individual, indivisible operations, called transactions."

"Each transaction must succeed or fail as a complete unit; it cannot remain in an intermediate state.

"Transaction mandatorily requires acknowledgment to get received as a necessary feedback for accomplishment."

Why that's useful (by way of example)

Current Account:

£2000

Savings Account:

£5000

£500

?

Why that's useful (by way of example)

Current Account:

£2000

-£500

Savings Account:

£5000

+£500

Why that's useful (by way of example)

Current Account:

£1500

Savings Account:

£5000

+£500

A problem has been detected and Windows has been shut down to prevent damage to your computer.

PFN_LIST_CORRUPT

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:
*** STOP: 0x0000004e (0x00000099, 0x00900009, 0x00000900, 0x00000900)

Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further assistance.

Why that's useful (by way of example)

Current Account:

£1500

Savings Account:

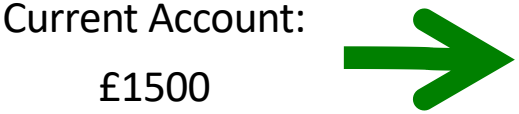
£5000

£500?

Why that's useful (by way of example)

Current Account:

£1500



Savings Account:

£5000

£500

Lets try that again only this time with transactionality!

Why that's useful (by way of example)

Current Account:

£1500

-£500

Savings Account:

£5000

+£500

Before we start... log the current state!

```
Transaction Log:
Start Transaction Transfer from Current to Savings. Current: £1500 Savings: £5000
```

Why that's useful (by way of example)

Current Account:

£1000

Savings Account:

£5000

+£500

```
Transaction Log:
Start Transaction Transfer from Current to Savings. Current: £1500 Savings: £5000
Deducted £500 from Current.                         Current: £1000 Savings: £5000
```

Why that's useful (by way of example)

Current Account:

£1000

Savings Account:

£5000

+£500

If the system crashes now we can roll back the transaction when we reboot, setting the accounts back to their original values!

```
Transaction Log:
Start Transaction Transfer from Current to Savings. Current: £1500 Savings: £5000
Deducted £500 from Current.                          Current: £1000 Savings: £5000
```

Why that's useful (by way of example)

Current Account:

£1000

Savings Account:

£5500

```
Transaction Log:
Start Transaction Transfer from Current to Savings. Current: £1500 Savings: £5000
Deducted £500 from Current.                         Current: £1000 Savings: £5000
Add £500 to Savings.                                Current: £1000 Savings: £5500
```

Why that's useful (by way of example)

Current Account:

£1000

Savings Account:

£5500

Finally log that the transaction is complete!

```
Transaction Log:
Start Transaction Transfer from Current to Savings. Current: £1500 Savings: £5000
Deducted £500 from Current.                         Current: £1000 Savings: £5000
Add £500 to Savings.                                Current: £1000 Savings: £5500
End Transaction Transfer from Current to Savings.   Current: £1000 Savings: £5500
```

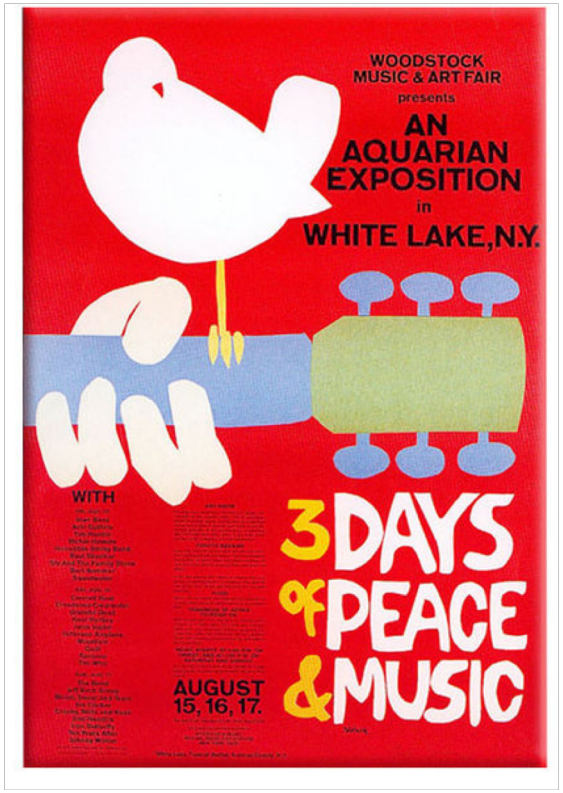
So Transactional Processing...

- Very, very useful
- Stops you losing money!
- Keeps your system in a consistent state
- Bit of a pain to code though...

1969

1969

Woodstock



1969

Woodstock

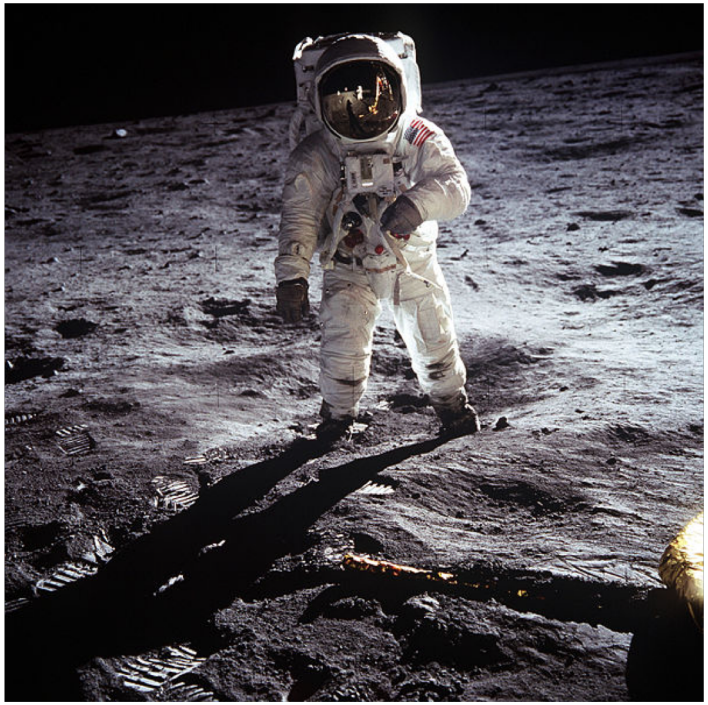
Maiden Flight of Concorde

1969

Woodstock

Maiden Flight of Concorde

Neil Armstrong walks on the moon



1969

Woodstock

Maiden Flight of Concorde

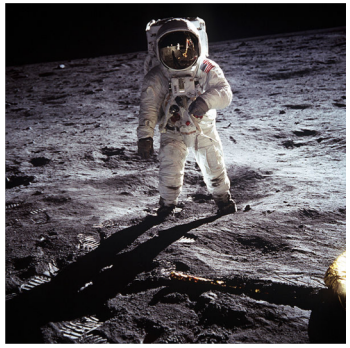
Neil Armstrong walks on the moon

The Beatles play their last live gig


WOODSTOCK
MUSIC & ART FAIR
presents
AN
AQUARIAN
EXPOSITION
in
WHITE LAKE, N.Y.
3 DAYS
of PEACE
& MUSIC
AUGUST
15, 16, 17.

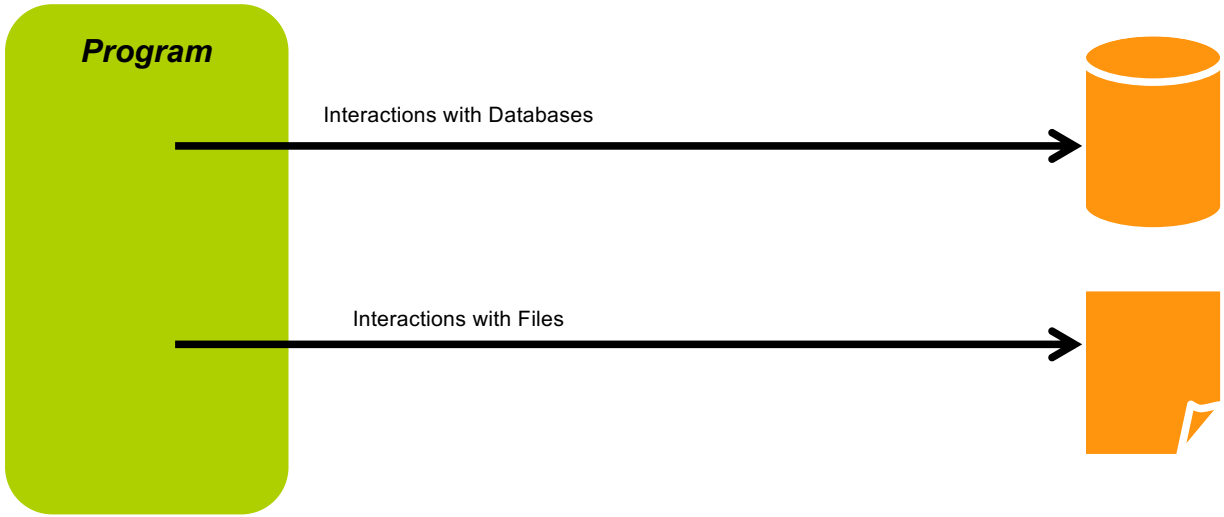
1969

Clearly with all this going on people didn't want to have to faff with coding transactionality into programs anymore!



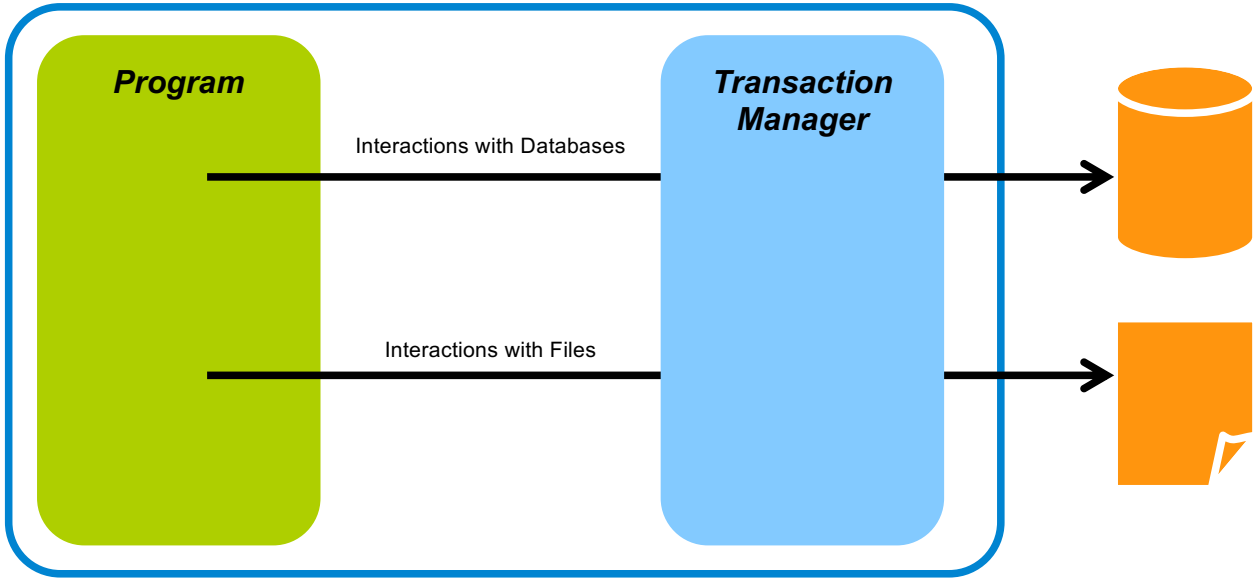
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



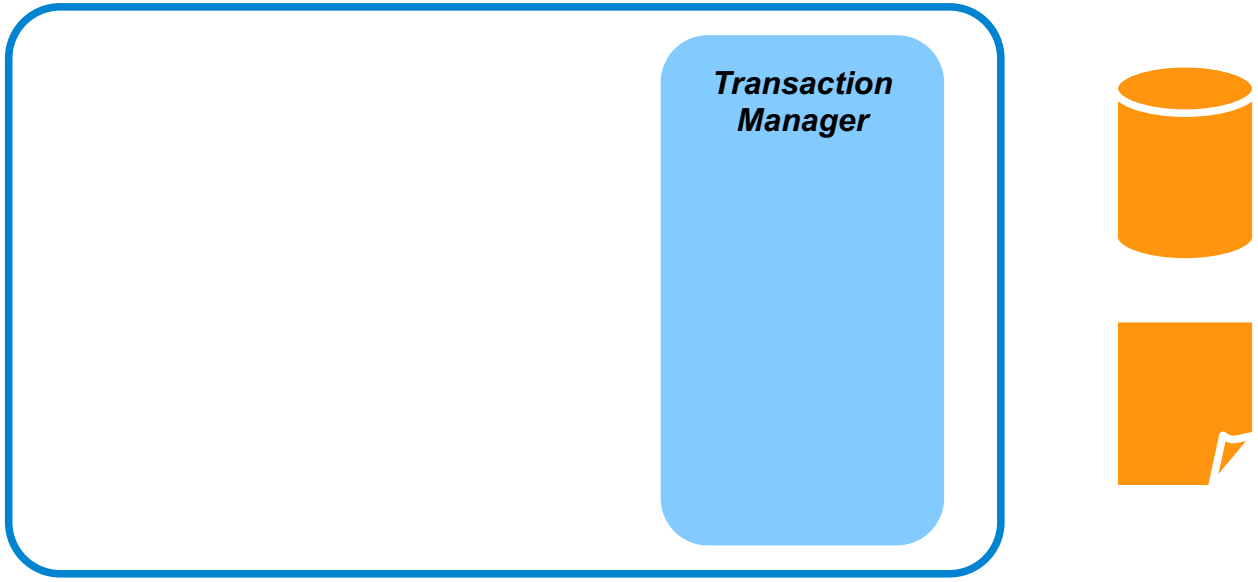
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



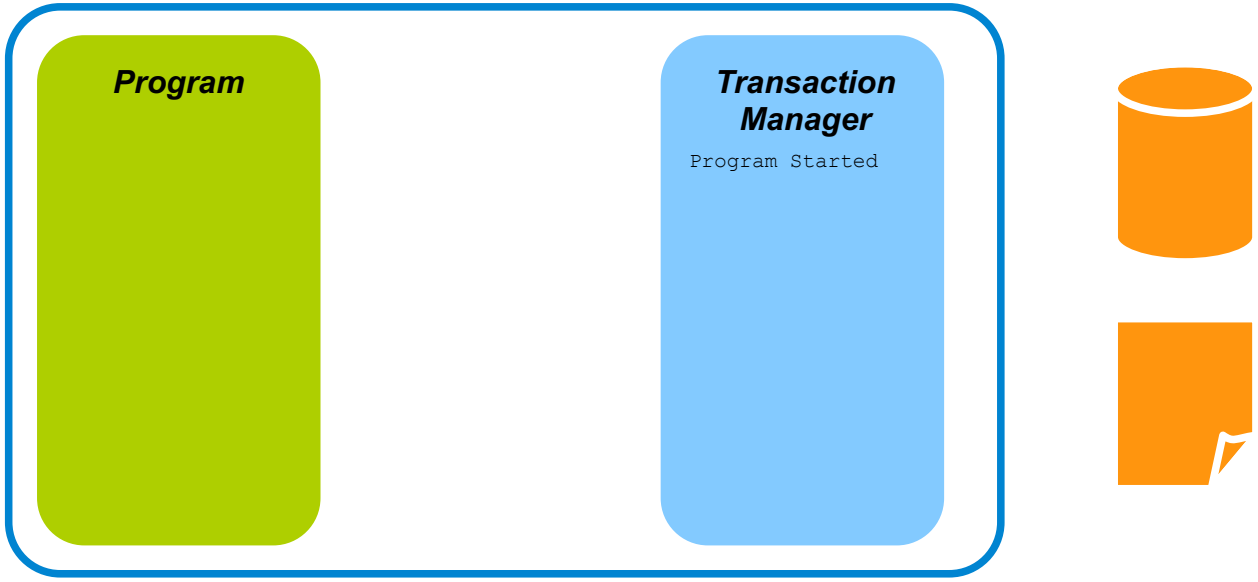
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



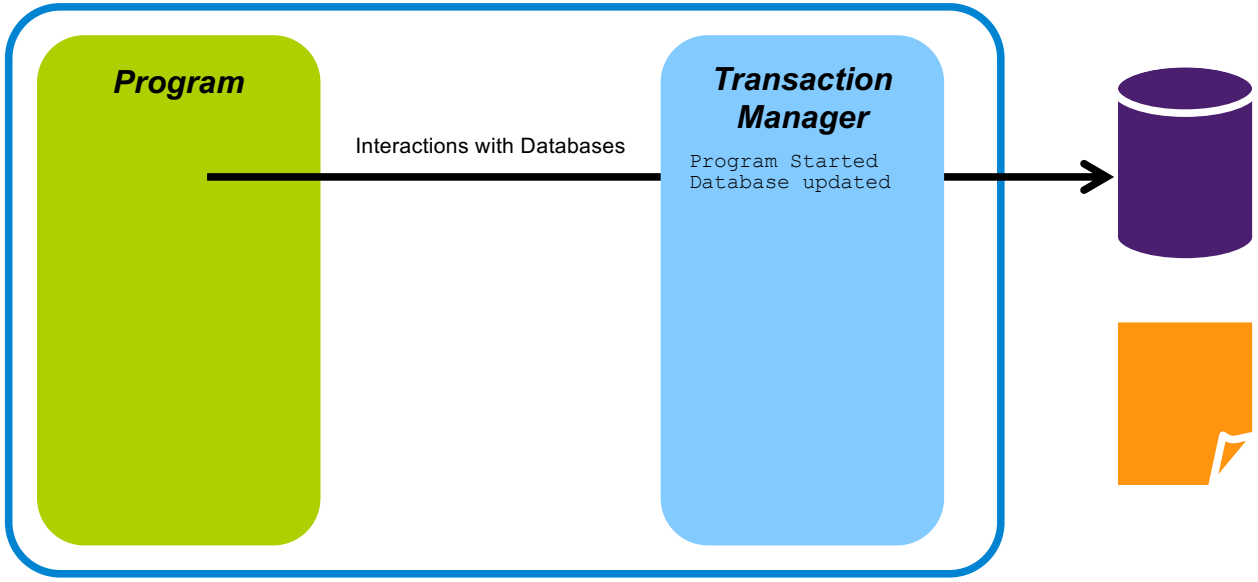
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



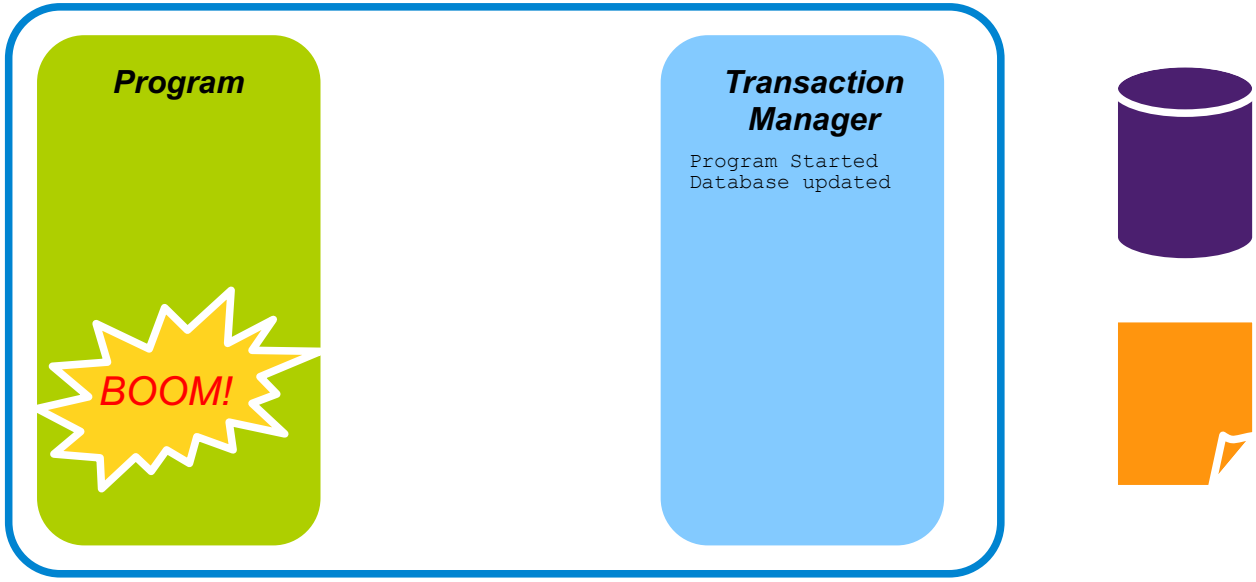
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



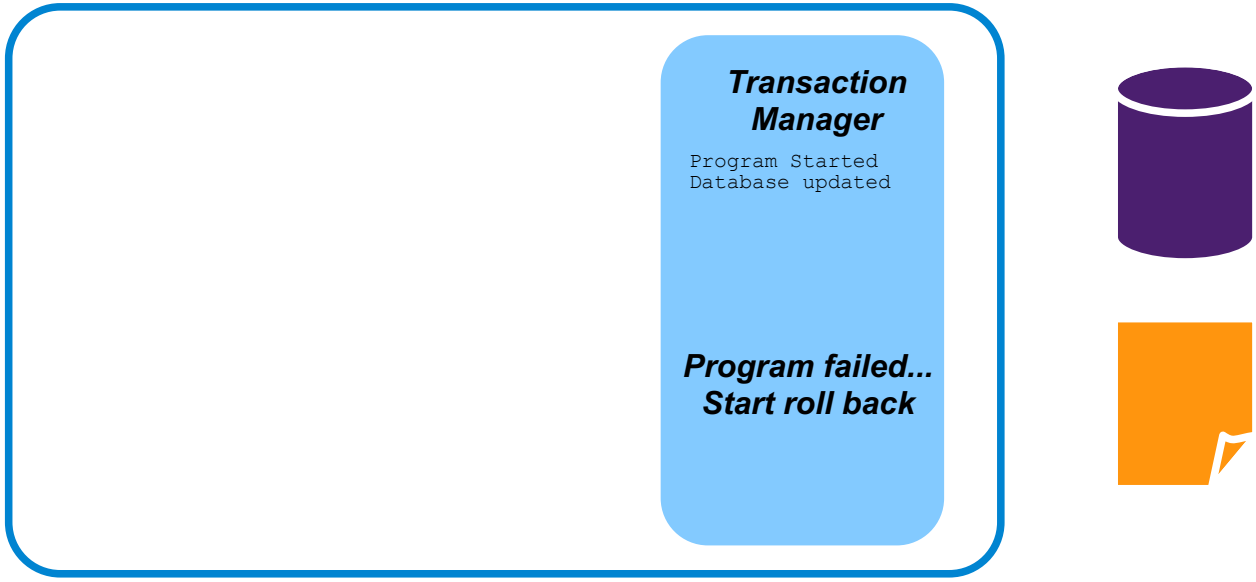
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



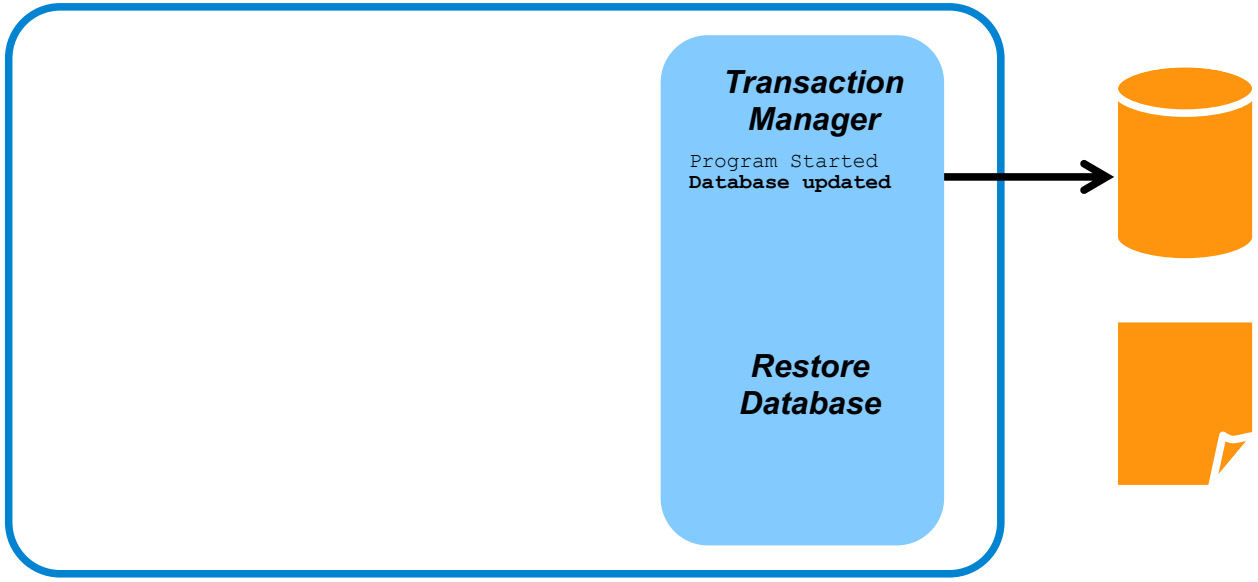
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



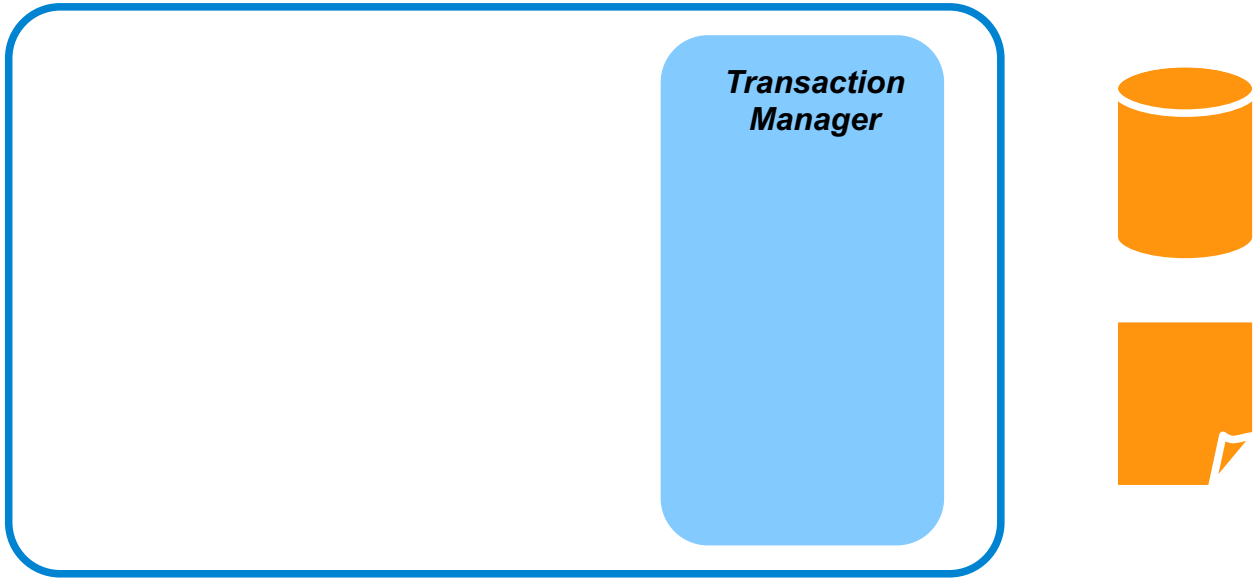
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



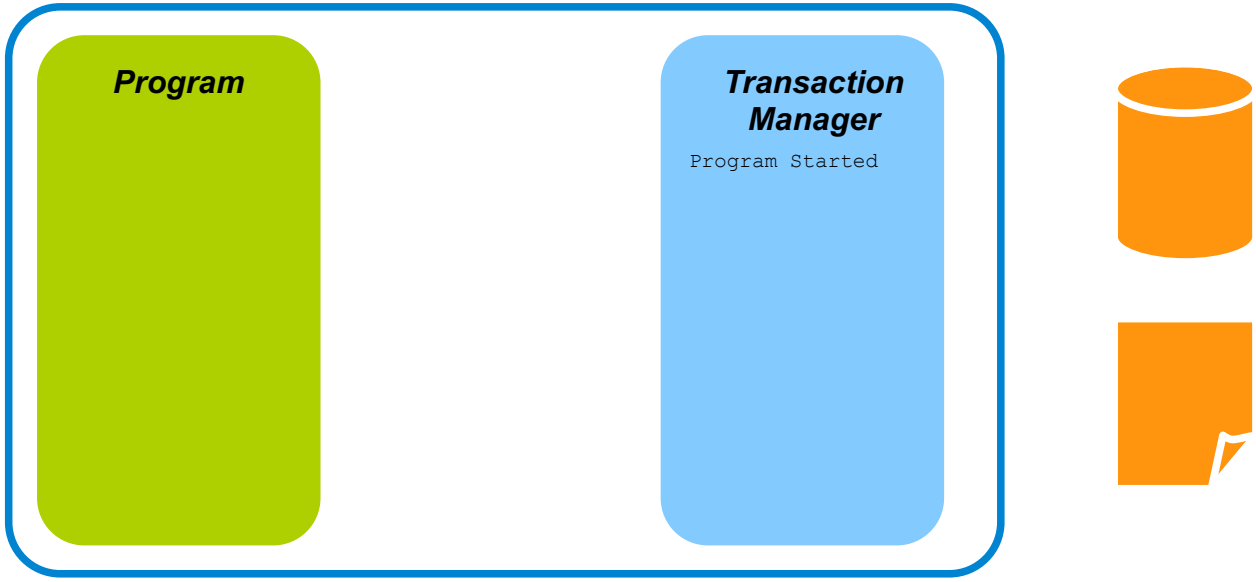
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



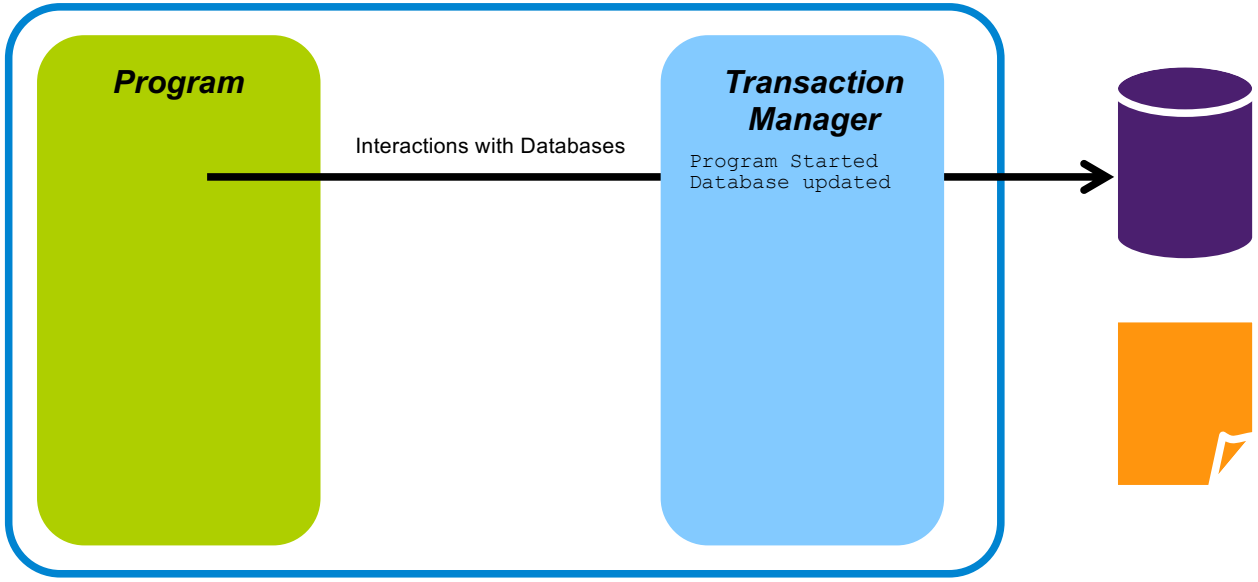
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



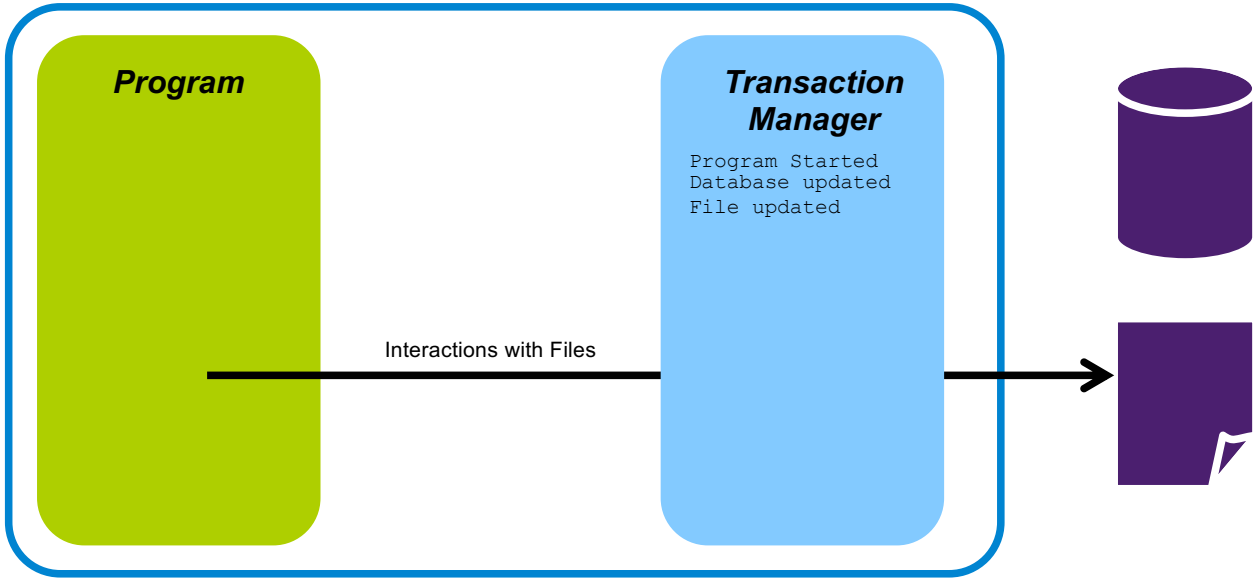
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



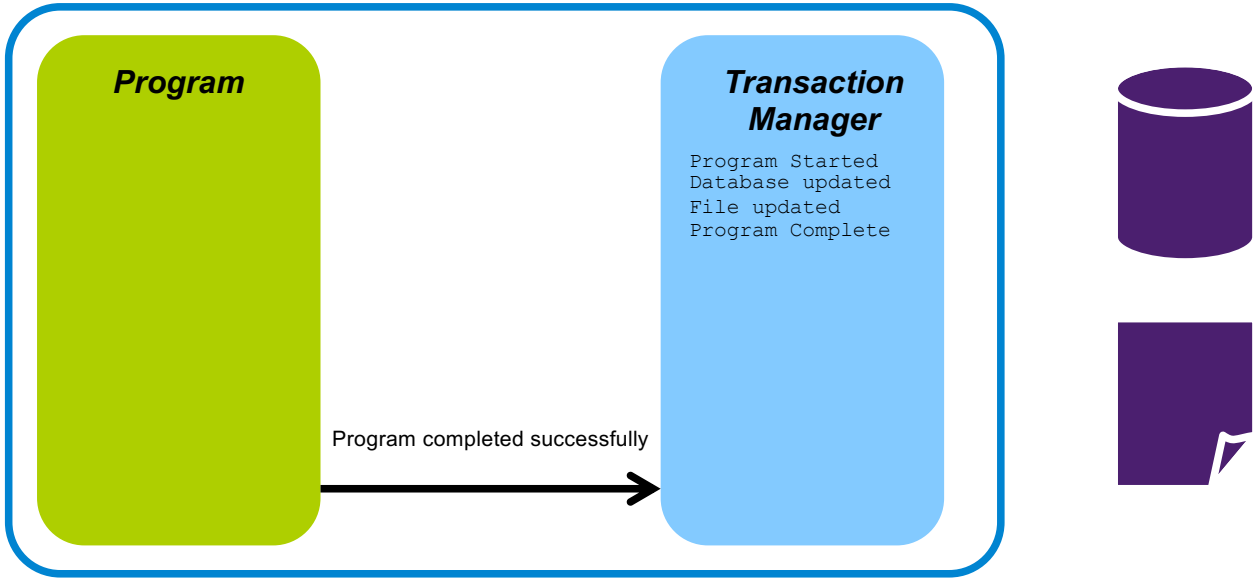
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



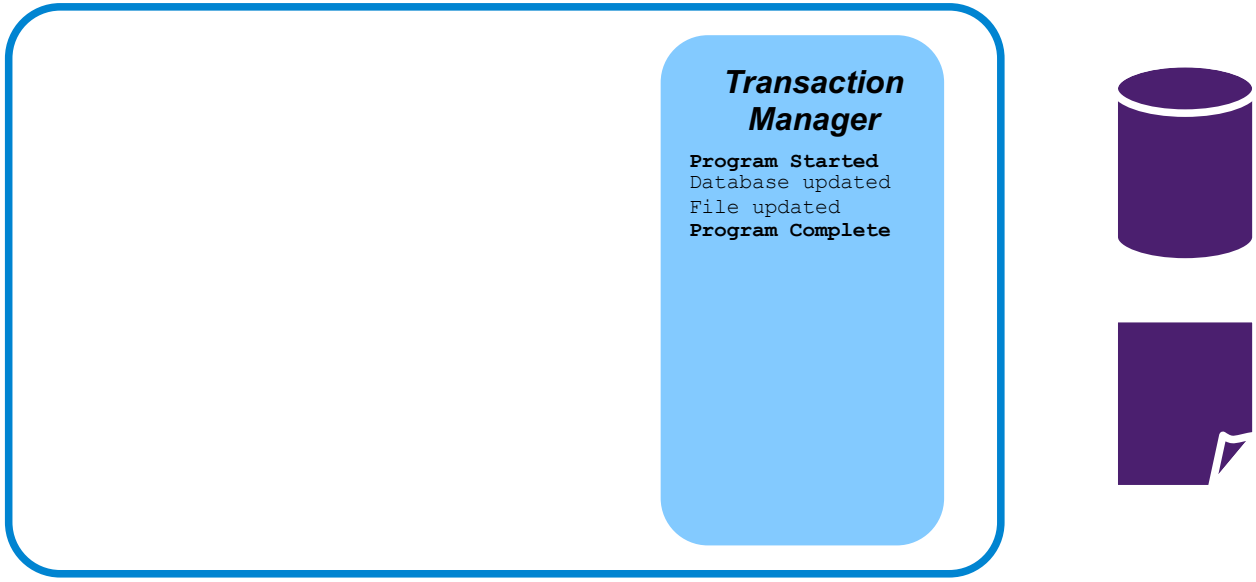
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



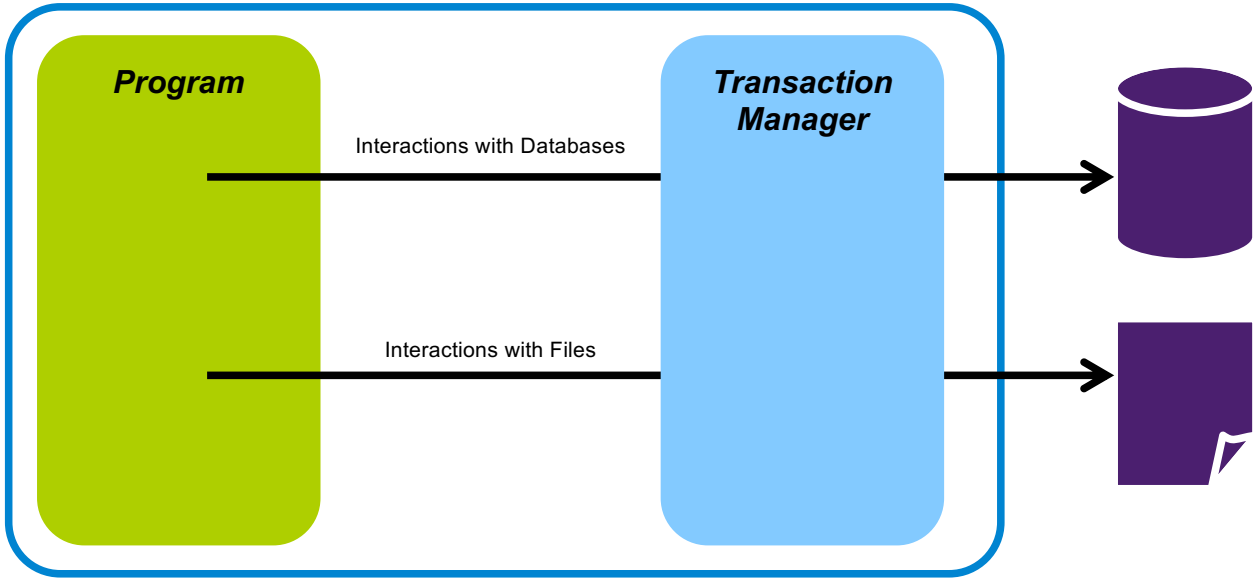
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



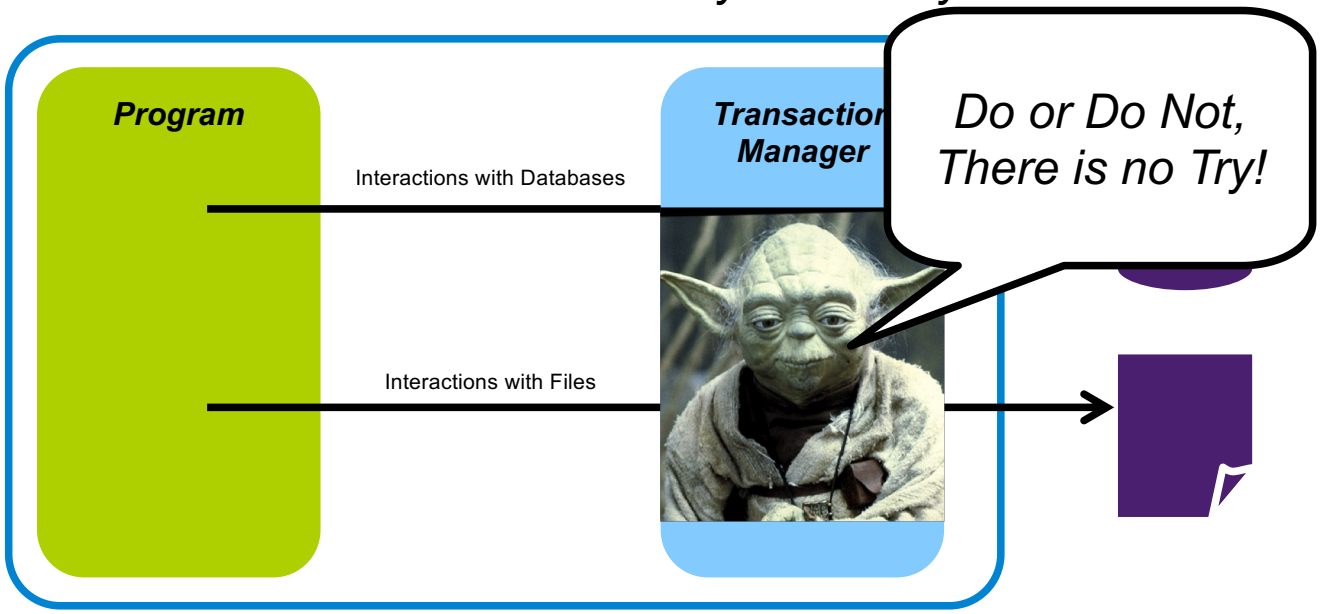
So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



So IBM had an idea...

What if a program could be run on a system which took care of the transactionality for it so that programmers didn't need to code it manually into every function?



1969

Woodstock

Maiden Flight of Concorde

Neil Armstrong walks on the moon

The Beatles play their last live gig

First release of CICS is available to buy!

Since then it has been hugely successful

"CICS is probably the most successful piece of software of all time . . . It is the mainstay of business computing throughout the world . . . Millions of users unknowingly activate CICS every day, and if it were to disappear the world economy would grind to a halt."

Phil Manchester, (Personal Computer Magazine, March 1994)

....and even more so today!

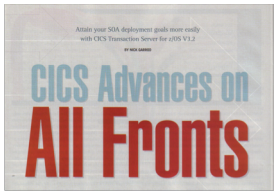
"Although most people are blissfully unaware of CICS, they probably make use of it several times a week, for almost every commercial electronic transaction they make. In the whole scheme of things, CICS is much more important than Microsoft Windows."

Martin Campbell-Kelly, From Airline Reservations to Sonic the Hedgehog

(A History of the Software Industry, MIT Press 2003)

CICS is one of the "35 Technologies that shaped the industry"

(Computerworld, Sept 2002)



Since then it has been hugely successful

The worlds business computing systems have evolved on CICS

CICS runs the businesses of over 90% of the Fortune 500 companies

CICS underpins many industries such as:

Banking
Insurance
Travel
Energy and Utilities
Retail
Telecoms
Governments


CICS®
Customer
Information
Control
System
ibm.com/cics
1:15 / 1:16

Keys to it's success

CICS is the best performing transaction server

Today can perform 200,000+ transactions per second

Billions of transactions per day world wide

How that compares to other internet traffic...

CICS is the most reliable transaction server

Z is for: Zero down time... CICS does not go down

CICS is the most secure transaction server

Core principles of security, governance and auditing built in

All possible because of System Z

A true beast of a machine

Can run multiple VMs both z/OS and Linux

Fit a warehouse size data center into one box

Spec:

Up to 170 Processors (5.2GHz)

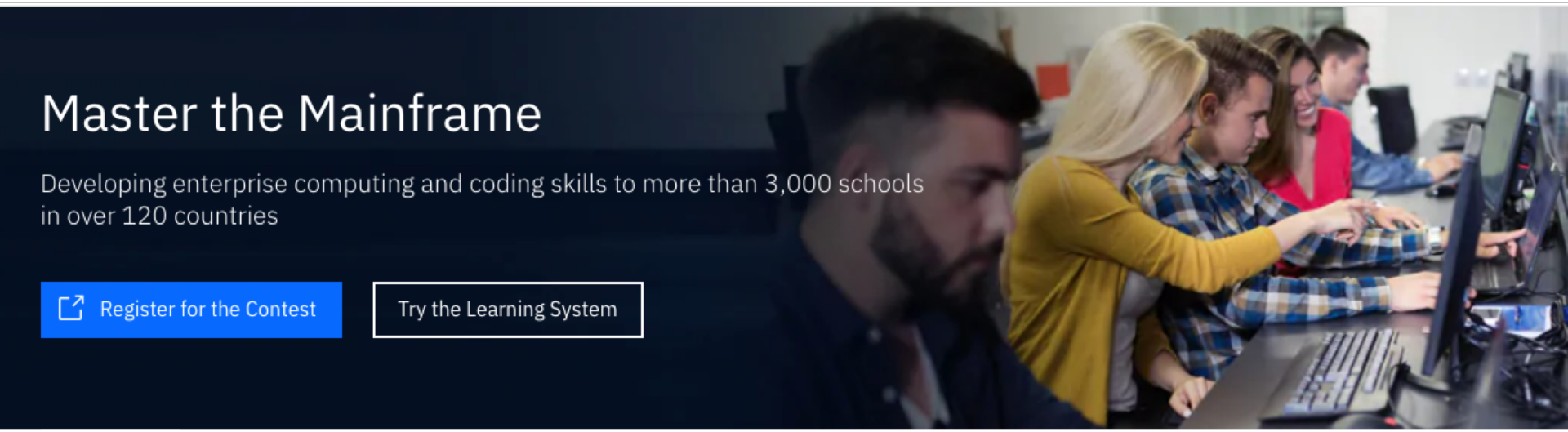
Up to 32 TB Memory

Massive Redundancy

No single point of failure

Optional water cooled

The worlds most expensive boiler!

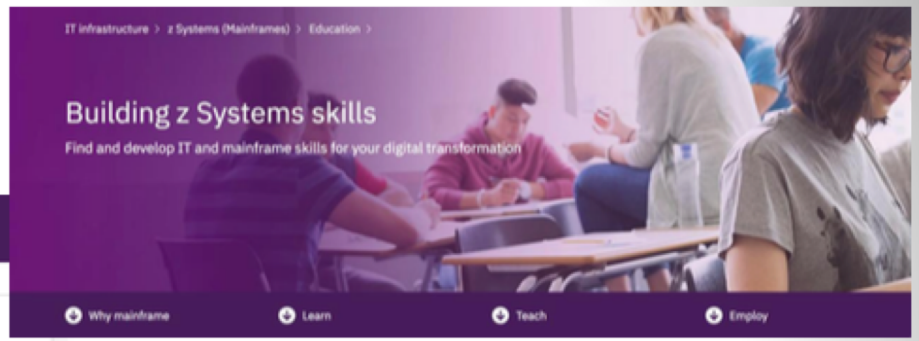


2018 Contest NOW OPEN

https://ibm.biz/masterthemainframe

Refreshed Z Skills Website

ibm.biz/zskills



Learn

Become a Mainframer
Mainframers share how they got started and what helped most (02:35)

Master the Mainframe
Start learning on virtual platform and earn open badges

Access free e-Learning
Enrolled students get 44 hours of online courses and earn open badges

Find a job
Got mainframe skills? Post your resume on the z Systems Job Connector

Join the community
Join the z Systems Student Advocacy Hub to learn and connect

Take a class
Find and attend a specialized class near you

Teach

Access courseware
Find presentations, labs, quizzes and more for over 25 courses

Access systems
Get free access to z/OS and Linux on z for hands on exercises

Promote your program
Email details of your available courses to share with employers and students

Use the guide
The Educator QuickStart offers everything you need to get started (425 KB)

Master the Mainframe
Incorporate the learning system or contest into your class. It's self-graded and available on demand.

Join the community
Join the z Systems Advocate Community to stay connected, learn and share

Employ

Find mainframe talent
Universities worldwide offer mainframe education and talent (266 KB)

Employer Reference Guide
Find resources to help you recruit, train and retain z Systems skills. (78 KB)

Post a job
Use the zJobconnector to find z Systems mainframe skills

Explore training
The Skills Gateway provides learning journeys and access to technical training for your employees

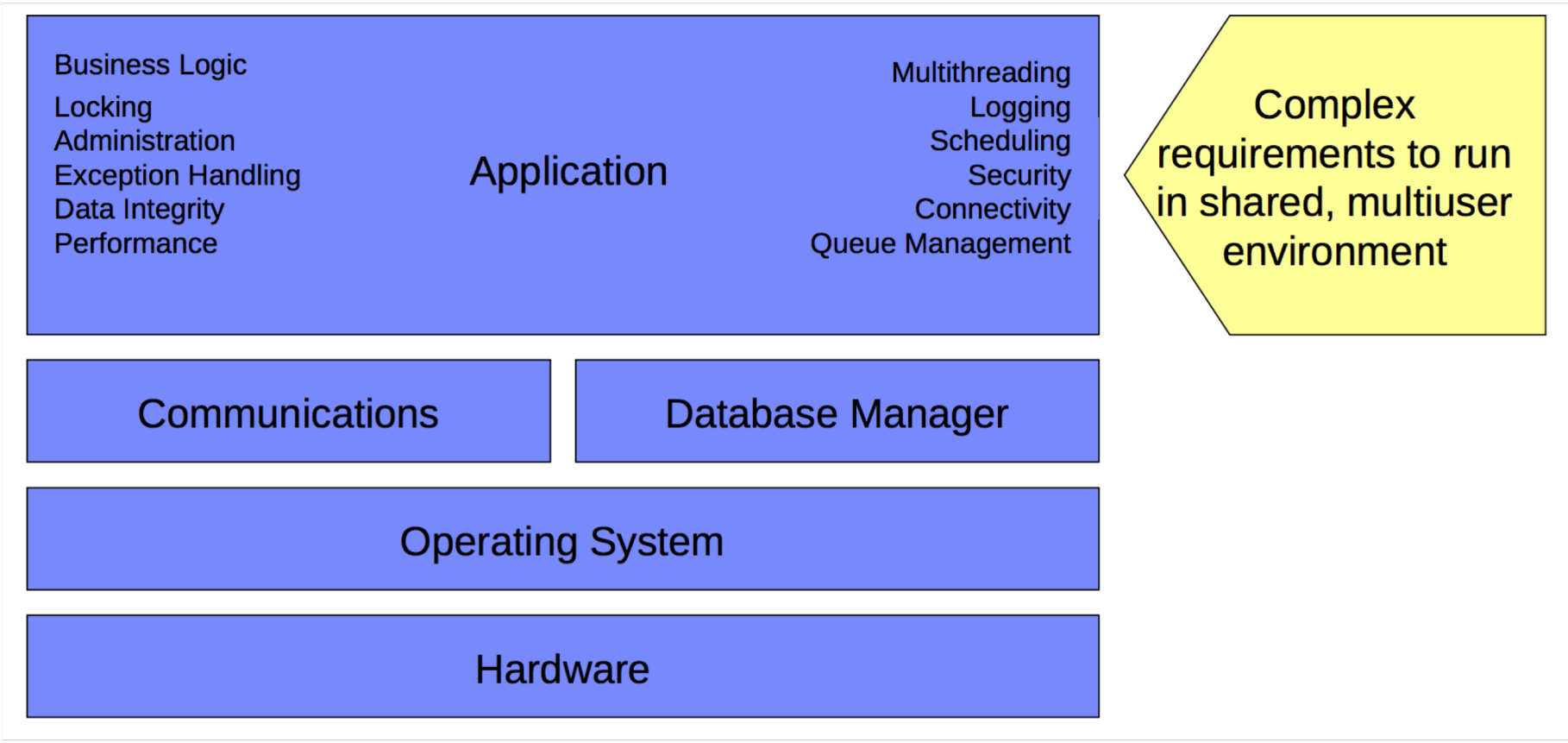
Join the community
Join the z Systems Employer Advocacy Hub to engage local students, learn and connect

Gen z programs
Connect your young professional mainframers through Gen z programs

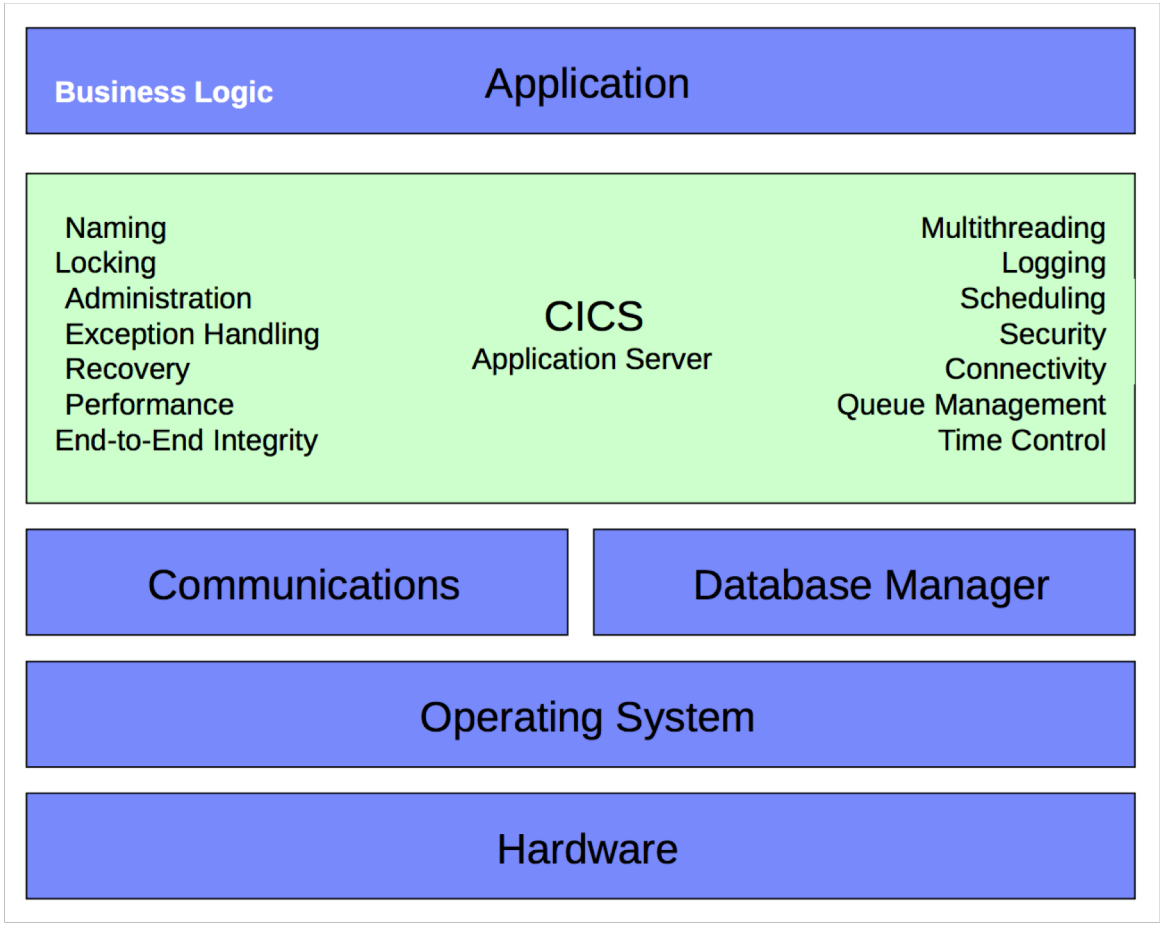
Your Client's One-Stop Skill Shop!

14

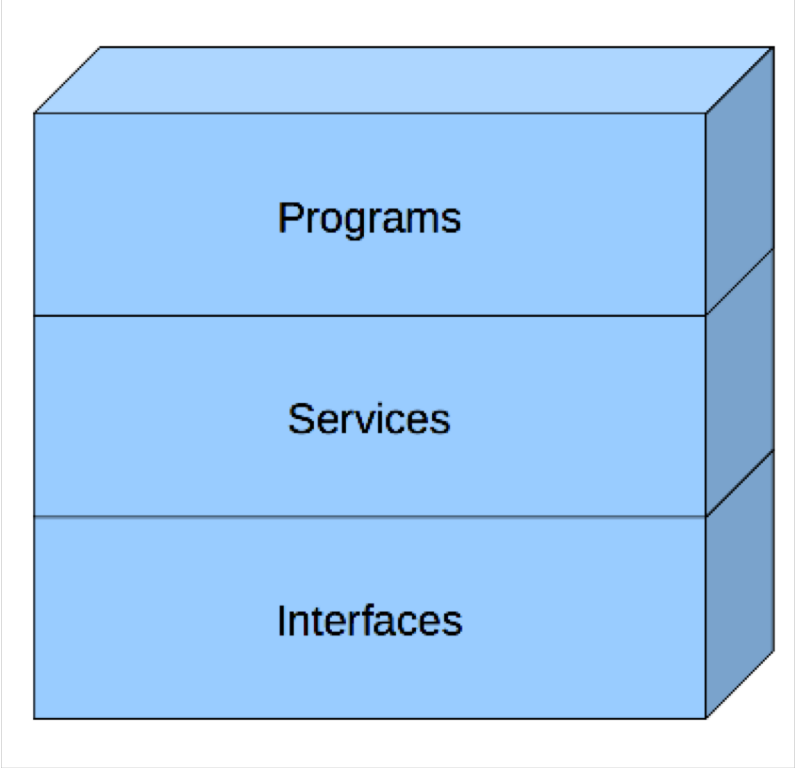
More than just transactions…



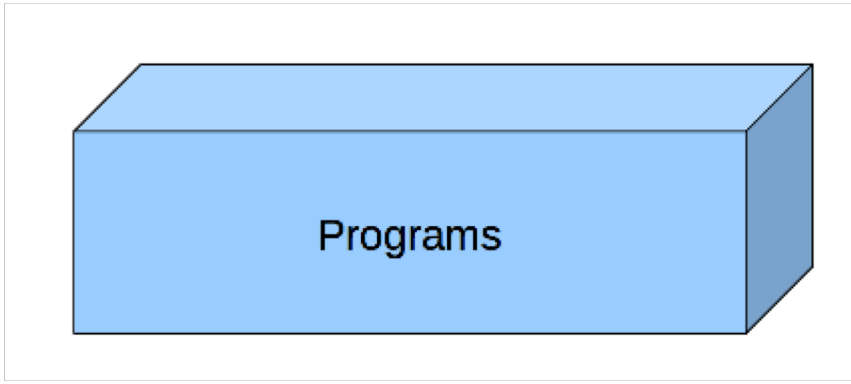
More than just transactions…



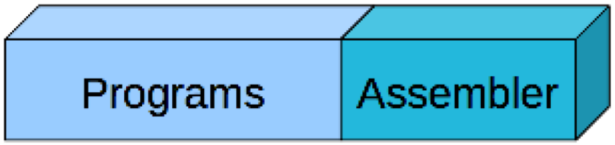
CICS - A Mixed Language Application Server



Programs



Programs - Assembler




```
CH1       DC    CL8'BLEEDING‘
CH2       DC    CL4'EDGE'
CICS      DS    CL12
*
          MVC   CICS(8),CH1
          MVC   CICS+8(4),CH2
```

Works on physical machine instructions

See “Principles of Operation” for more information :)

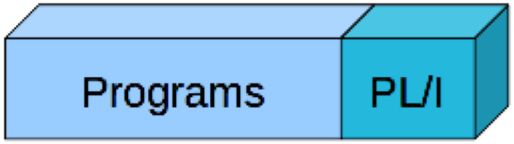
Programs - COBOL


```
01  BLEEDING  pic x(8)  value 'bleeding'.
01  EDGE      pic x(4)  value 'edge'.
01  CICS      pic x(12).

add BLEEDING to EDGE giving CICS.
```

COmmon **B**usiness-**O**riented **L**anguage

Grace Hopper - “mother of the COBOL language” (FLOW-MATIC)

Programs - PL/I




```
DCL BLEEDING CHAR(8) INIT( 'bleeding' );
DCL EDGE     CHAR(4) INIT( 'edge' );
DCL CICS     CHAR(12);

CICS = BLEEDING + EDGE;
```

"Programming Language One"

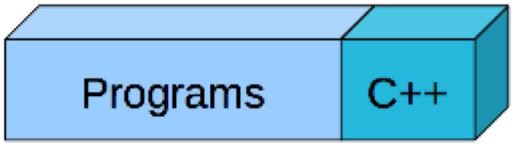
Originally called "New Programming Language"

Programs - C


```
char* BLEEDING = "bleeding";
char* EDGE     = "edge";
char* CICS;

main()
{
    CICS = strcat(BLEEDING, EDGE);
}
```

Named “C” because many of its features were derived from the “B” programming language

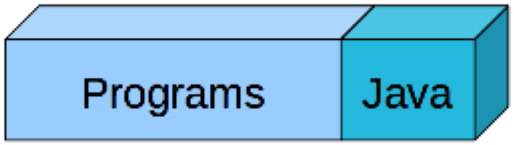
Programs - C++




```
char* BLEEDING = "bleeding";
char* EDGE     = "edge";
char* CICS;

main()
{
    CICS = strcat(BLEEDING, EDGE);
}
```

"new C" -> "C with Classes" -> "C++"

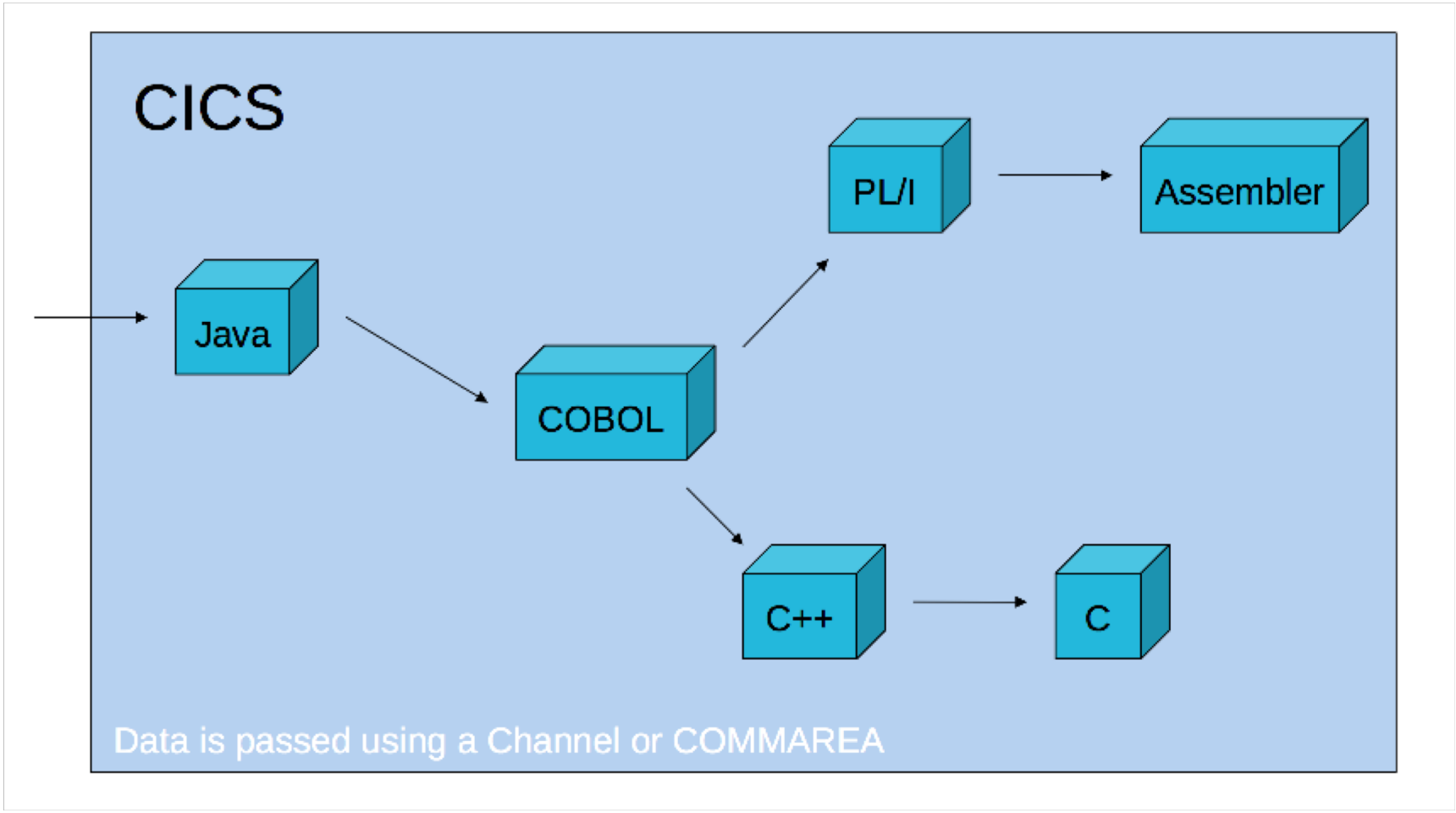
Programs - Java



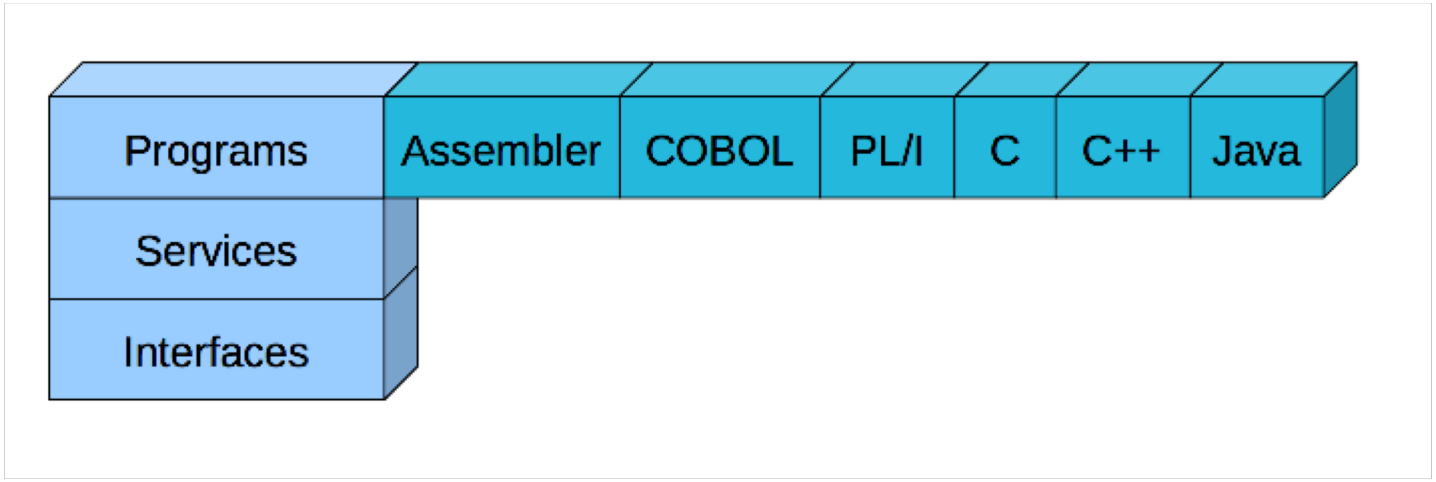

```
final String BLEEDING = "bleeding";
final String EDGE     = "edge";

String CICS = BLEEDING + EDGE
```

"Write once, run anywhere"

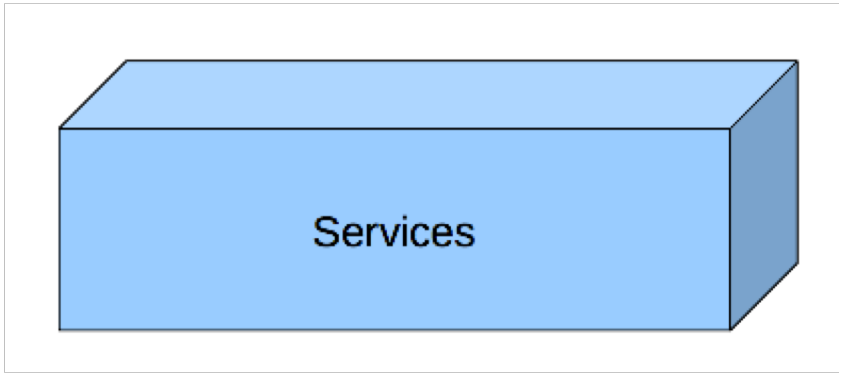
CICS Dynamic Program Linking



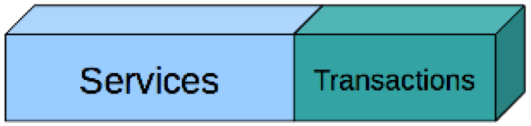
CICS - A Mixed Language Application Server

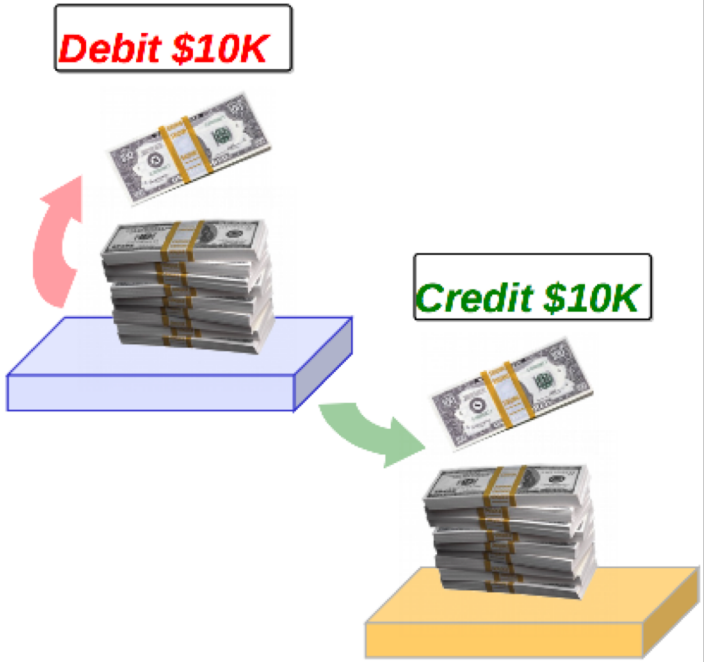


Services



Services - Transactions



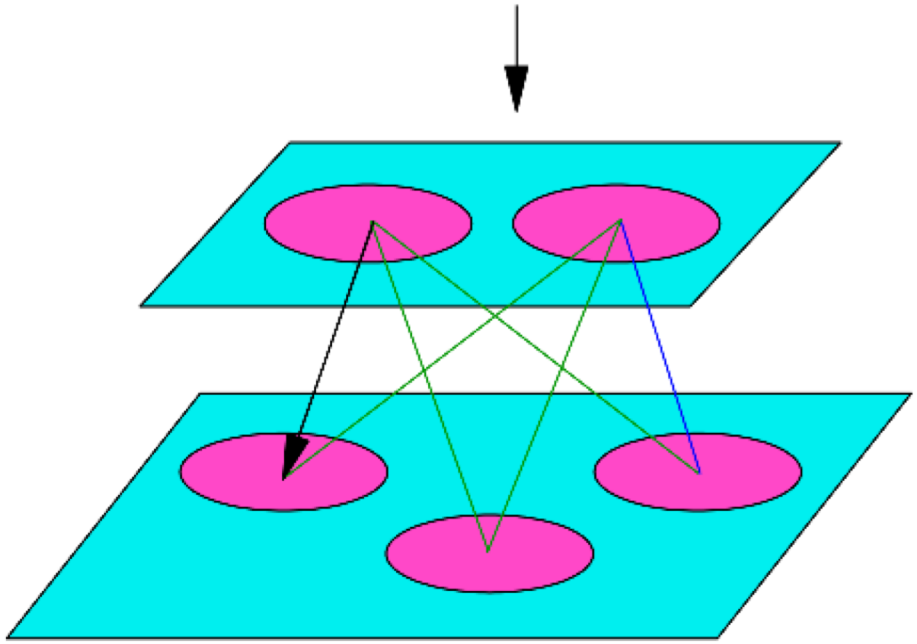


- Both operations must succeed (or fail) exactly once
  - Otherwise money goes missing!
- Operation:
  - Create unit of work
  - Perform update
  - Commit
  - or roll back if error

For a transaction manager to pass the ACID test it should provide:
Atomicity, Consistency, Isolation and Durability

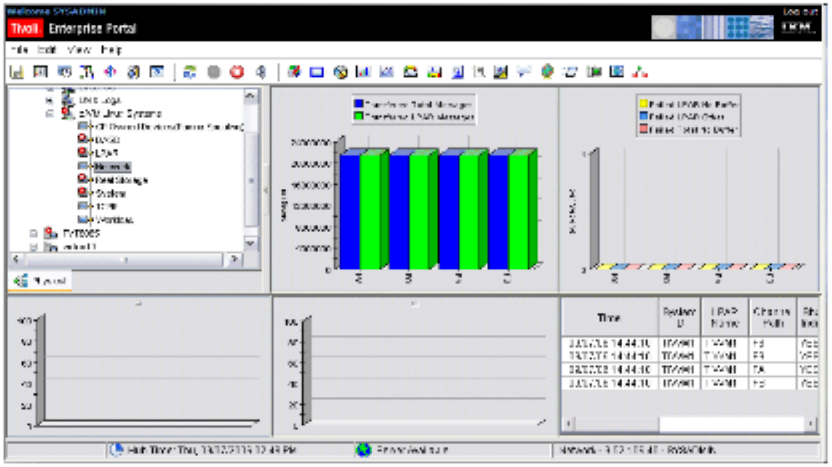
Services - Workload Management

- Dynamically balance workload across systems
- Balanced according to:
  - Queue algorithm
  - Goal algorithm



Services - Monitoring

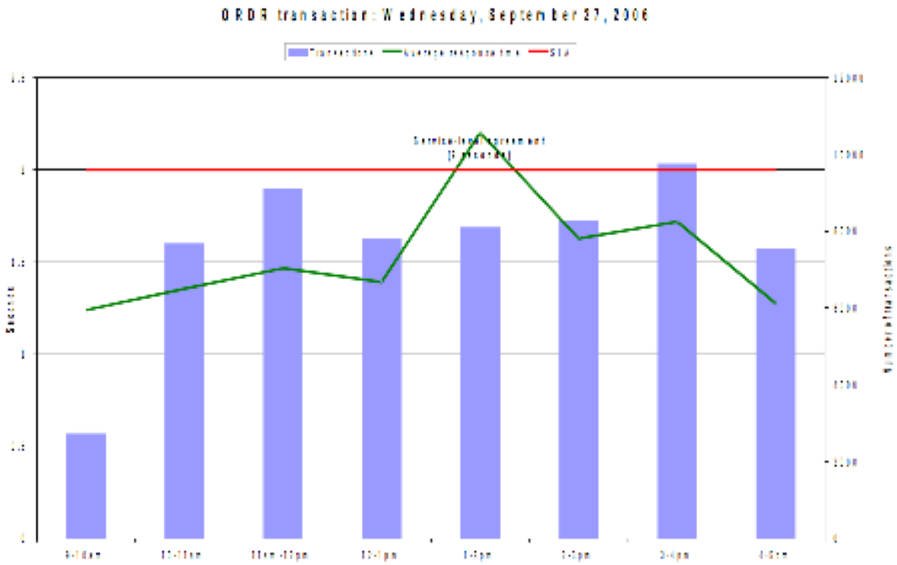
- Modern business require 24/7 availability
  - "living machines"
- Need ability to monitor
  - Resource usage
  - Resource availability



Screenshot of Tivoli Omegamon XE for CICS on z/OS

Services - Statistics

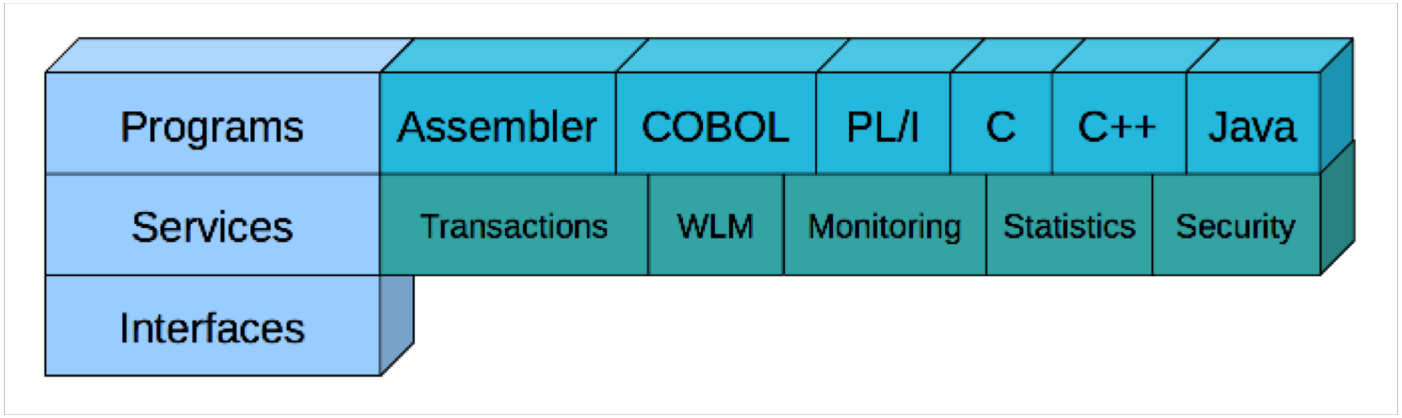
- Performance metrics are very important to 24/7 systems
- We want to know
  - How many times applications are run
  - Their response times
  - Bottlenecks



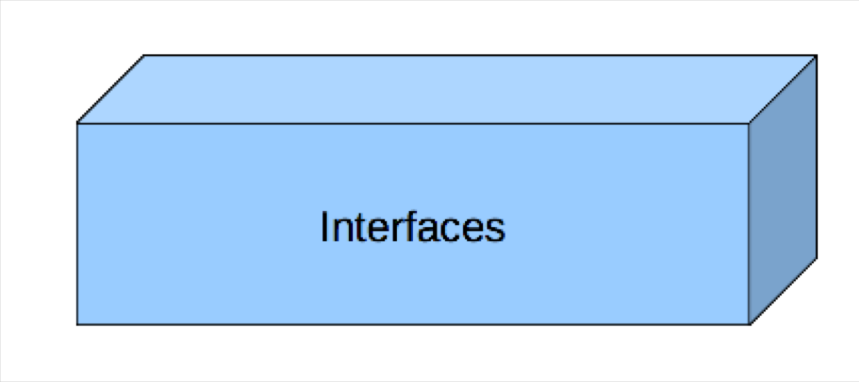
Services - Security

- Information is valuable and sensitive
- Trusted by customer to protect information
  - Credit card
  - Address
- An absolute must for companies so must be easy to implement

CICS - A Mixed Language Application Server

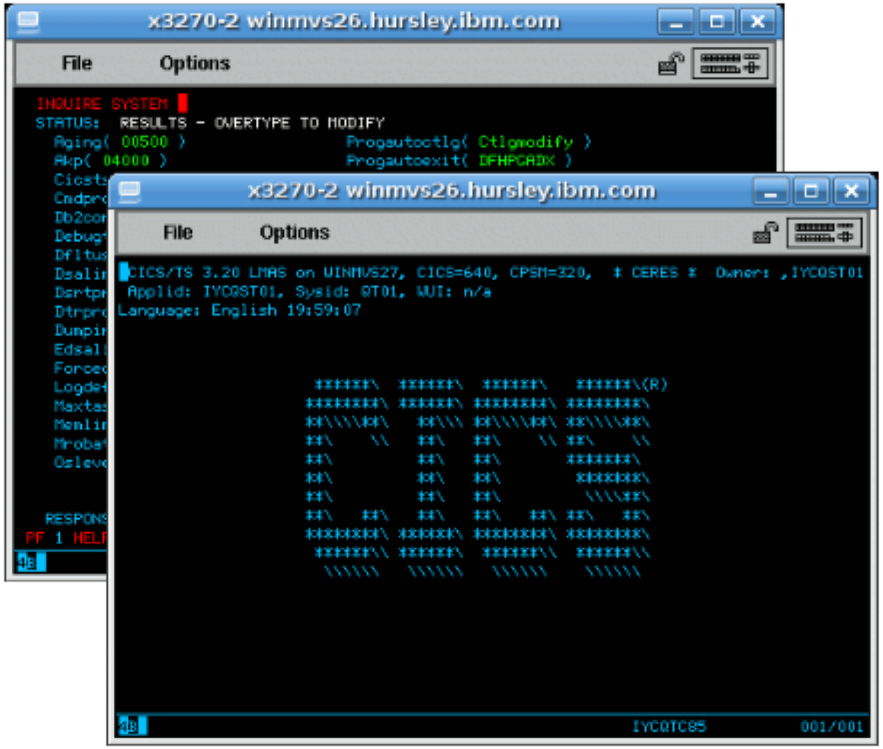


Interfaces



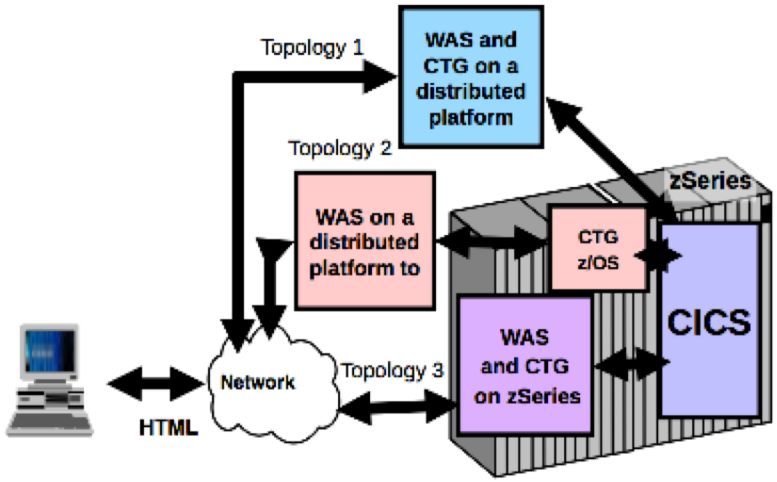
Interfaces - 3270

- One of the first display interfaces
- 24x80 characters
- Lovingly called “green screens”
- 3270 emulation



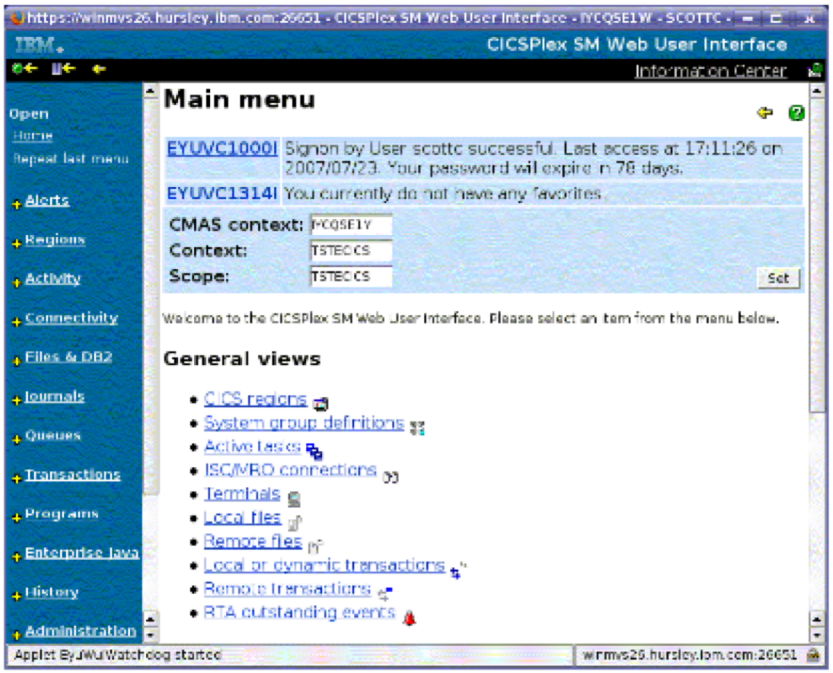
Interfaces - CTG

- IBM's popular connector from WebSphere to CICS
- Provides common interface to CICS from multi-platforms



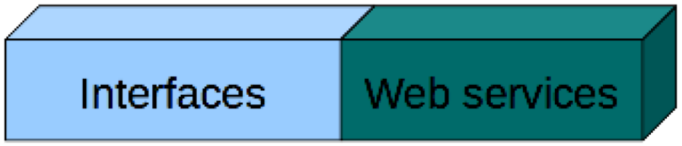
Interfaces - Web

- Users able to drive applications from anywhere
- Intuitive user interface displays
- Can be used by customers and administrators



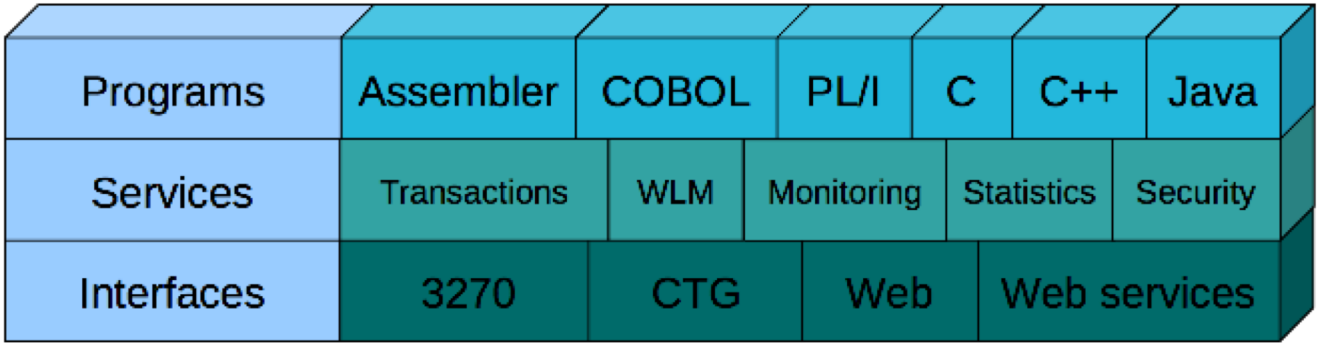
Screenshot of CICS Plex System Manager WUI

Interfaces - Web Services

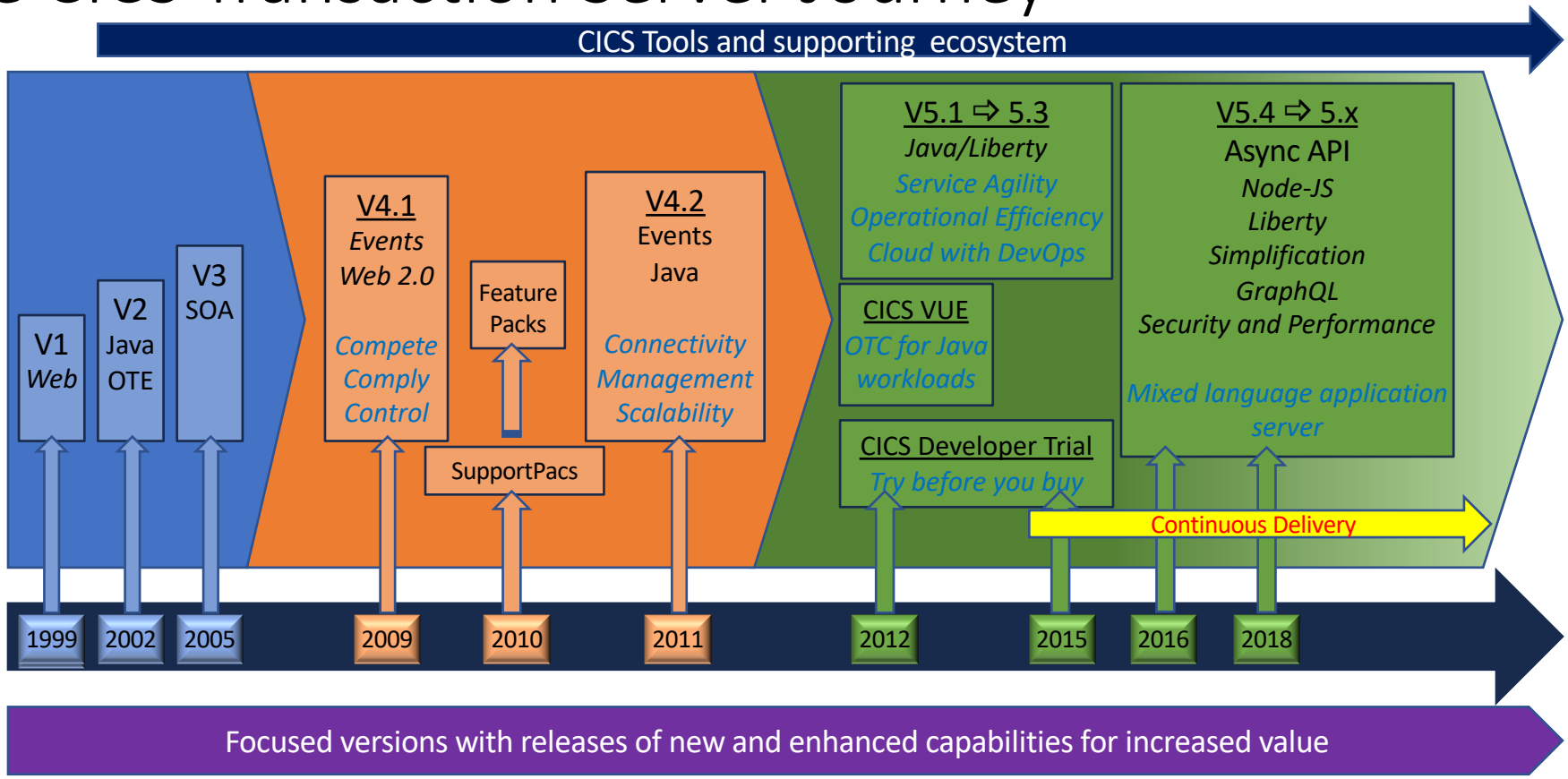


- "Anything talks to anyone"
- The latest and greatest
  - Based on open standards
    - XML, WSDL, Soap
- Platform and language agnostic
- WS-* standards for qualities of service and interoperability

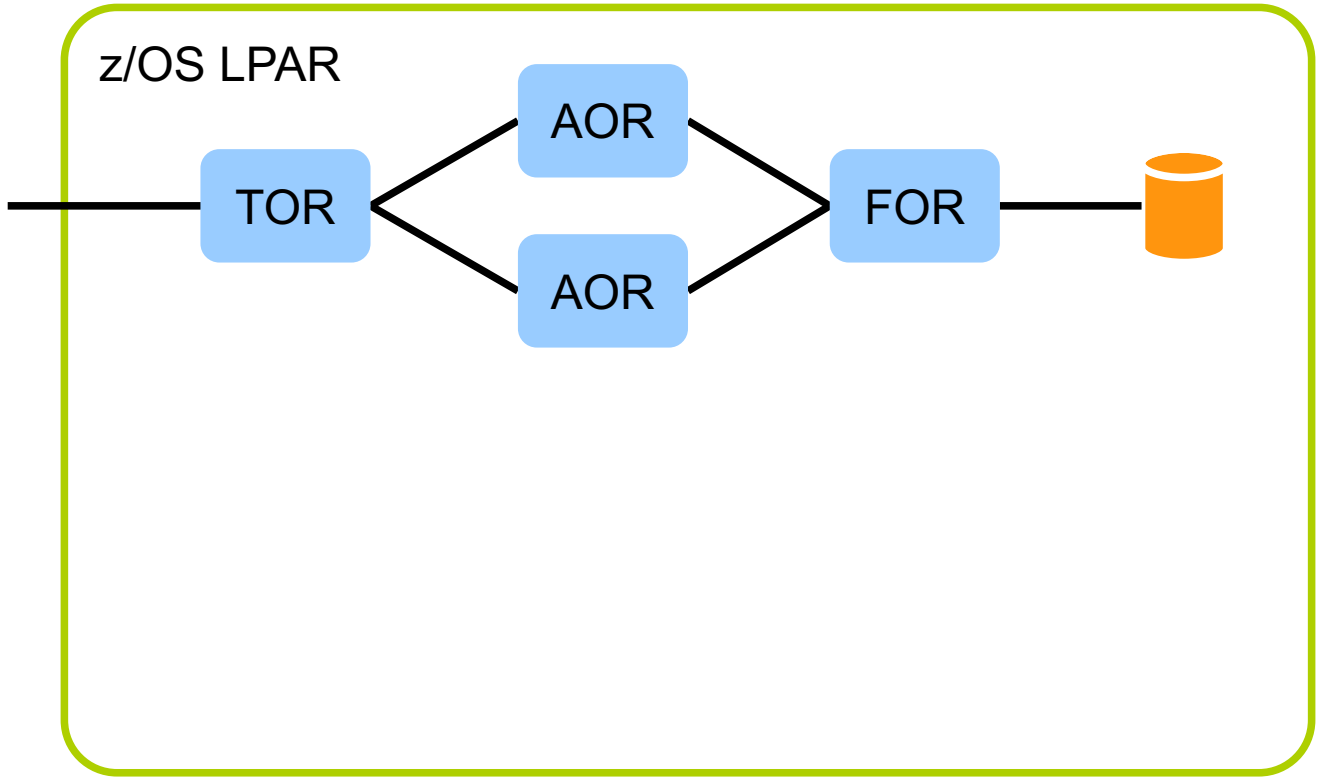
CICS - A Mixed Language Application Server



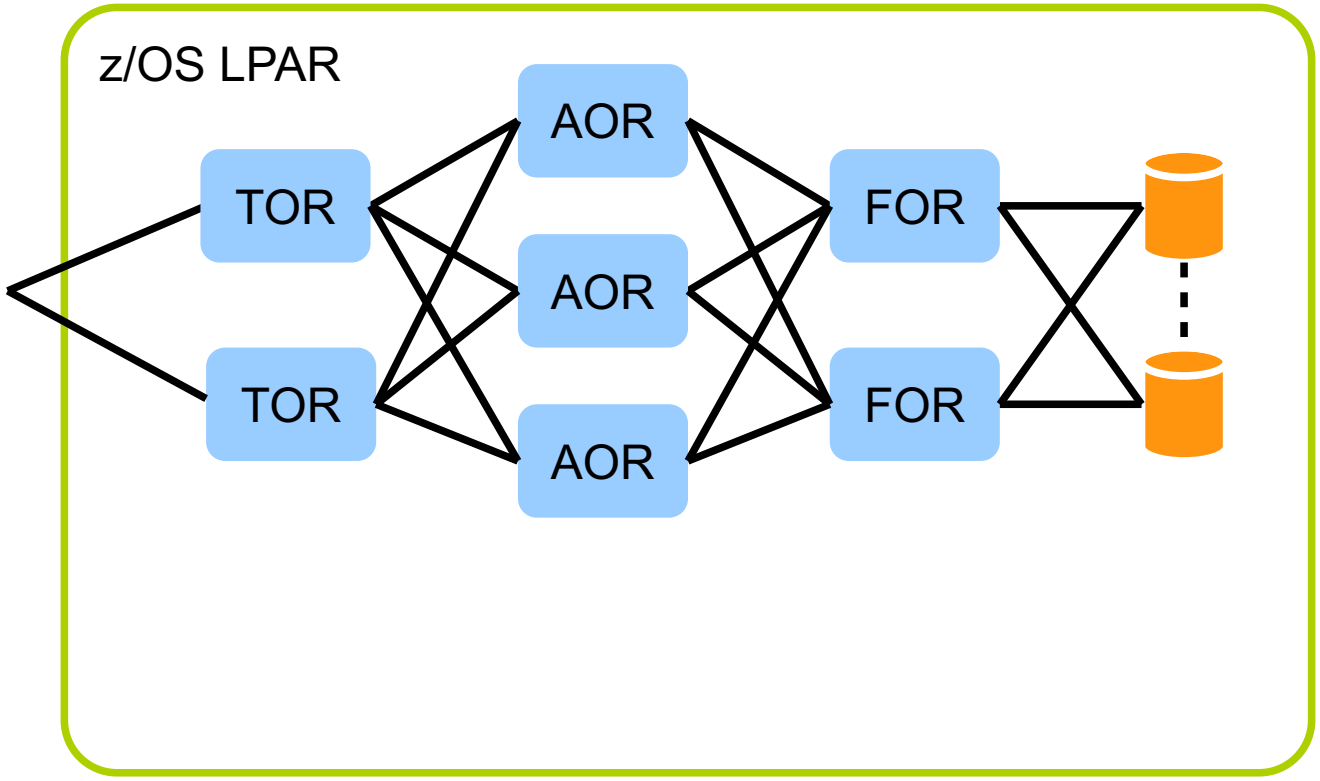
The CICS Transaction Server Journey



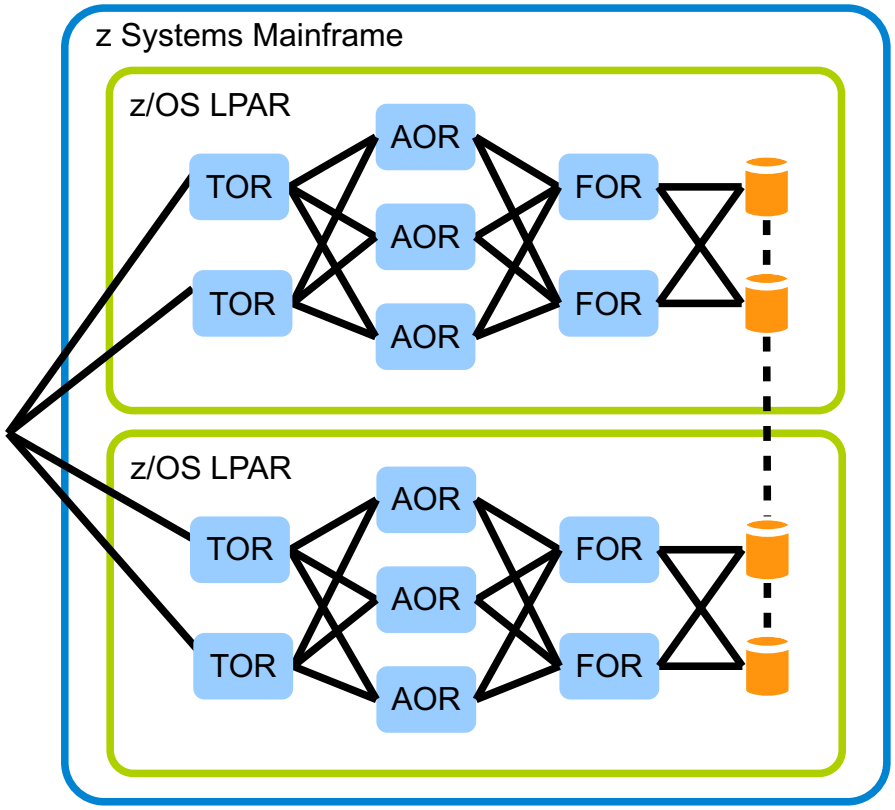
A Typical CICS Deployment



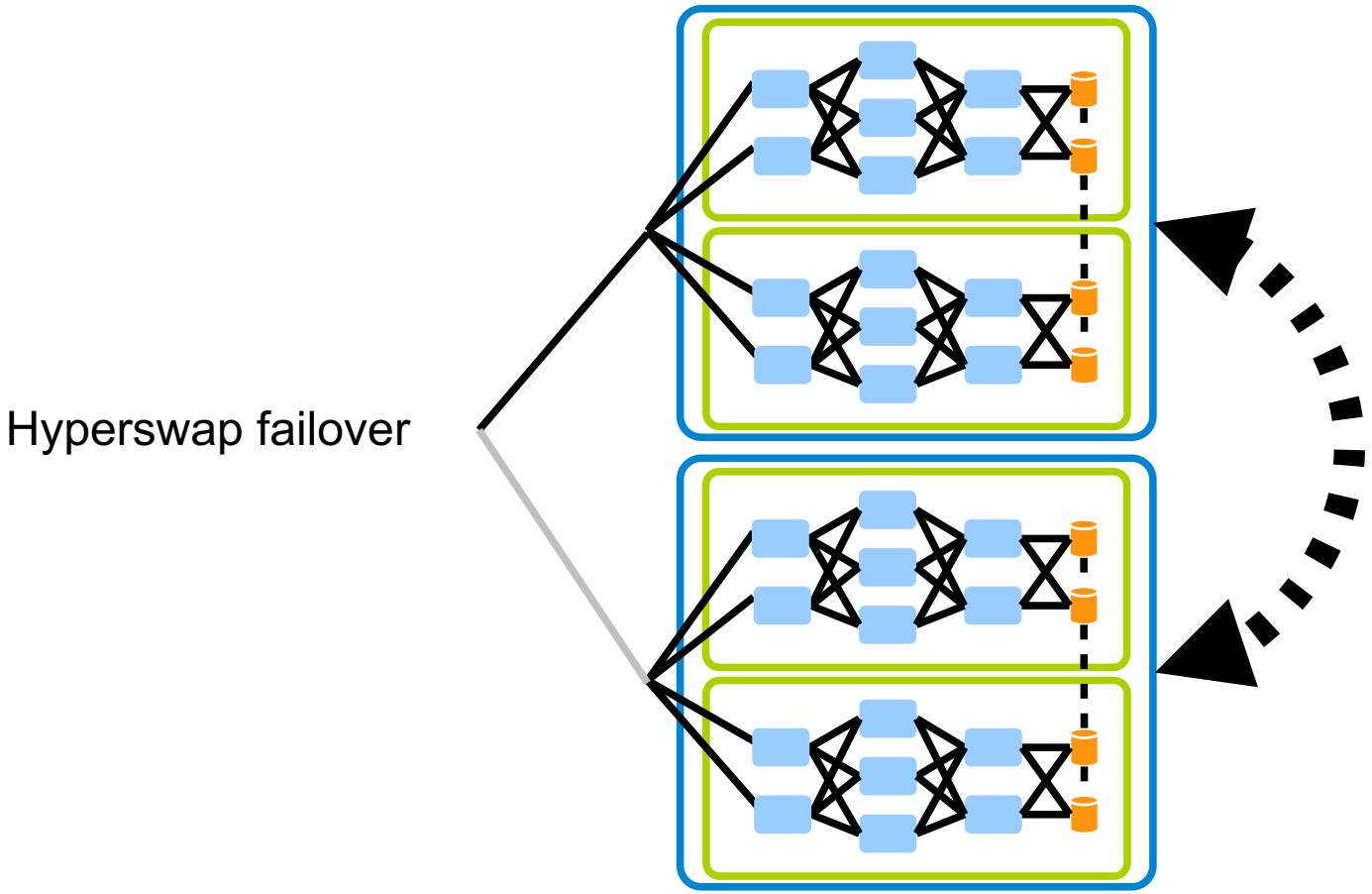
A Typical CICS Deployment



A Typical CICS Deployment



A Typical CICS Deployment



Where is CICS Developed?

IBM Hursley Labs near Winchester has been entrusted with CICS Development since 1974

Modern CICS Development

CICS has been in almost constant development for over 40 years

37 of those years were at Hursley

One of the oldest software products in the world still in use

Originally written in Assembler

Now mainly written in PLX and Java with some Assembler, C, and COBOL

Modern CICS Development

CICS has always embraced new practices and technologies

We were one of the first products in IBM to adopt Agile and Design Thinking

Many new capabilities and features have been added over the years

(and we are always adding more to support the cutting edge trends, APIs, Blockchain, Analytics etc)

CICS today can run programs written in:

Assembler, PL1, COBOL, C, C++, Java, and it even supports PHP and has a JEE web server!

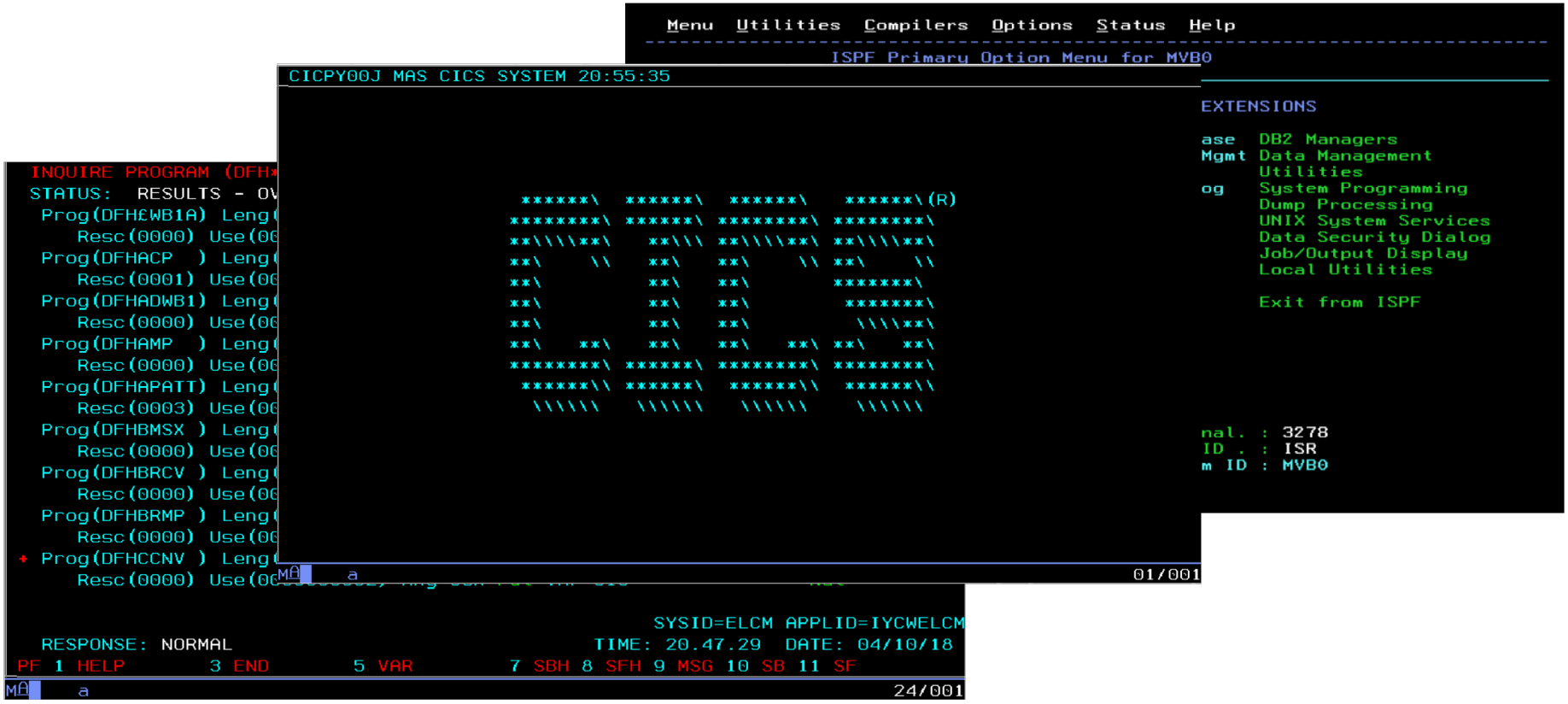
We are always looking to add more...

All programs are governed by the core rules of transactionality, security etc

Can easily link between programs of different languages

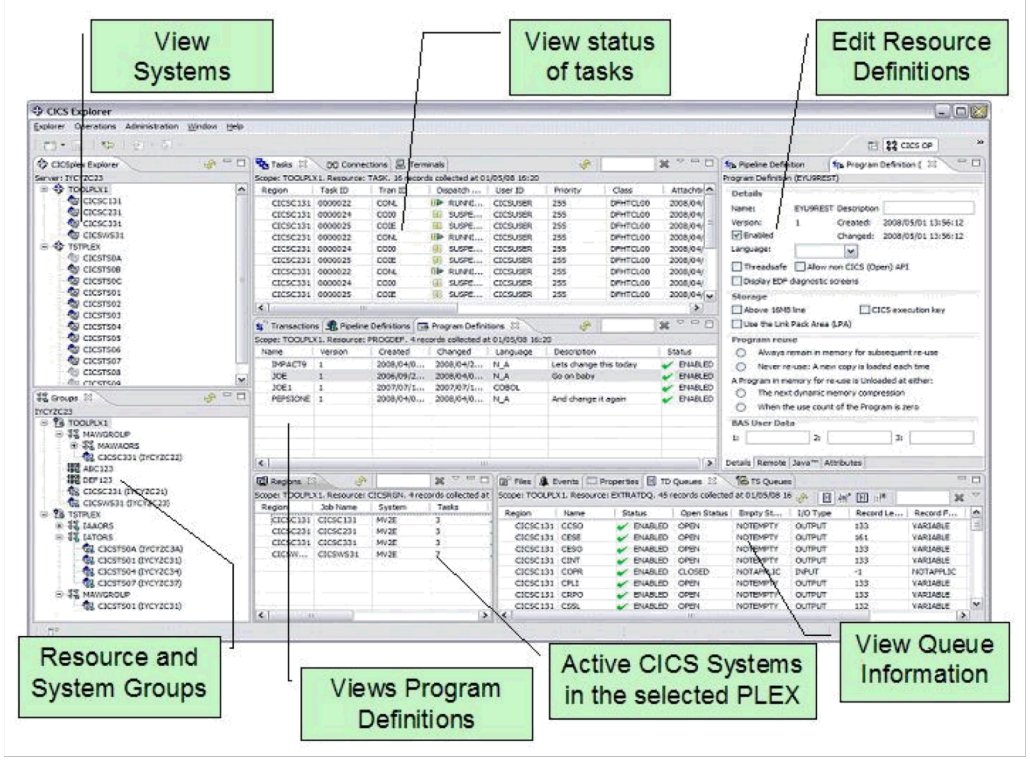
Interface Moderisation

When I joined CICS in 2002 the interfaces to it looked like this:



Interface Moderisation

Today it looks like this thanks to the CICS Explorer!



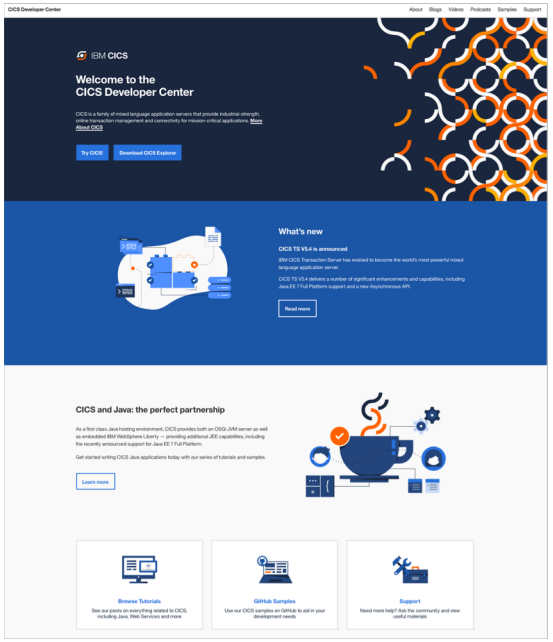
CICS Developer Center

IBM CICS Transaction Server V5.4

Continuous Delivery of CICS TS also requires Continuous Delivery of education

The CICS Developer Center has a number of resources to help users make the most of CICS :

- Blogs – around 120 technical articles to date
- Samples – hosted on GitHub
- Support – Q&A forums
- Podcasts, videos, client success stories & more



https://developer.ibm.com/cics

CICS Performance Series

IBM CICS Transaction Server V5.4

A series of educational videos covering the key topics and considerations for understanding CICS performance.

Including:

- Making sense of MIPs, MSUs and SUs
- LPAR capping
- CICS & z/OS WLM

These videos and our our IBM Redbooks publications can be found on one page in the Developer Center, here:

https://developer.ibm.com/cics/cics-performance-resources/

Getting Started with Java in CICS

IBM CICS Transaction Server V5.4

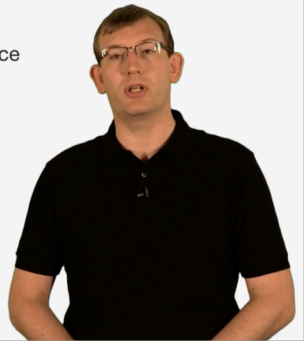
IBM CICS

Video course series from IBM Redbooks

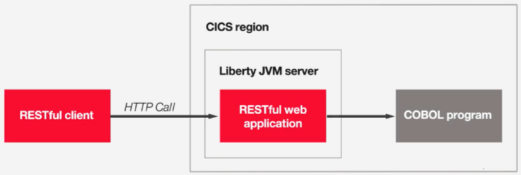
- Developing a RESTful Web application for Liberty in CICS
- Architecting Java solutions for CICS
- Extending a CICS web application using JCICS

What you'll learn by the end of this course

1. Developing a RESTful Java web service
2. Using the CICS Java API
3. Deployment of web applications



What you'll see in this course



https://ibm.biz/cics-java-courses

Introduction to CICS

IBM CICS Transaction Server V5.4

http://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0303.html?Open

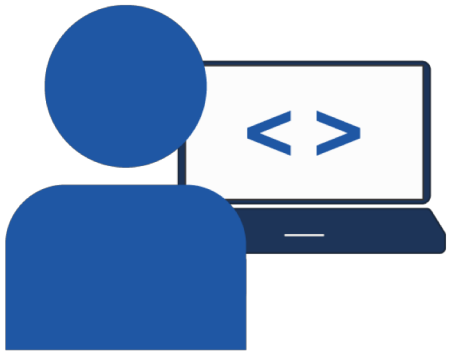
CICS TS V5.4 Developer Trial

IBM CICS Transaction Server V5.4

Try before you buy
- No charge license, no single version charging period

Feature rich for evaluation
- Some restrictions – 30 max tasks, works for 90+ days from download date

Supported
- Assistance via dwAnswers and normal IBM service
- APARs delivered in periodic service refresh
- See technote for details



Evaluate the value

Explore the capability

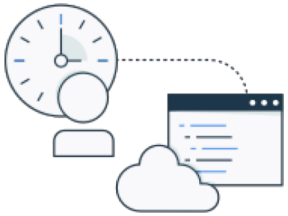
Create the business case

Order from IBM Shopz (as often as needed)

IBM z Systems Trial Program

IBM CICS Transaction Server V5.4

Experience the value of the latest IBM z Systems capabilities today at zero charge, and with no installation required.



No charge, on-demand environment

At zero cost and with short lead times, trying out the latest z Systems software is now easier than ever.

No setup, no install

Trial environments are pre-configured, and ready for use. Get started with your chosen offering in hours, not days.
Powered by z Development & Test.

Hands-on tutorials

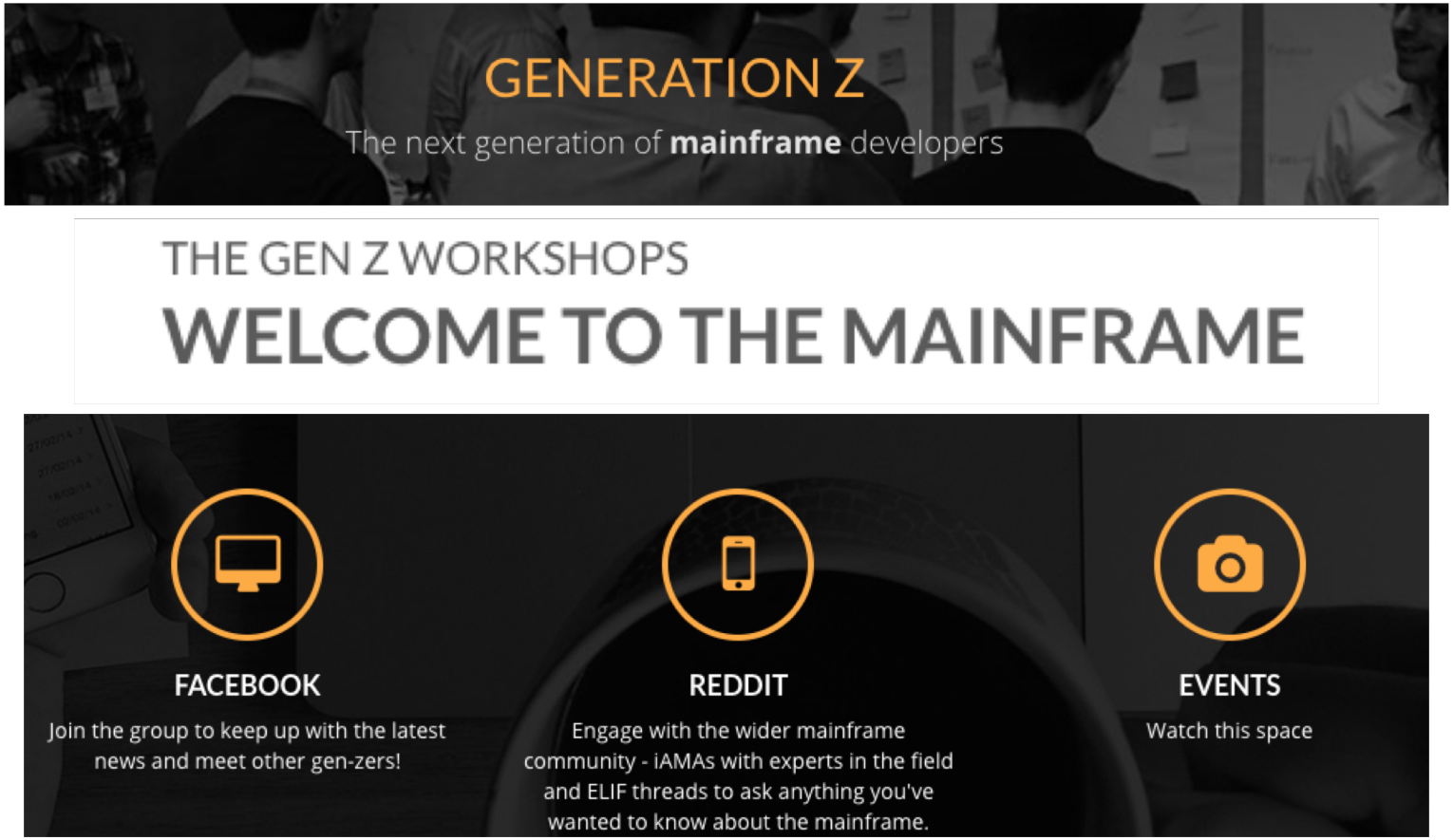
Short, easy-to-follow walkthroughs are included, so you'll experience your chosen product at its best in no time.

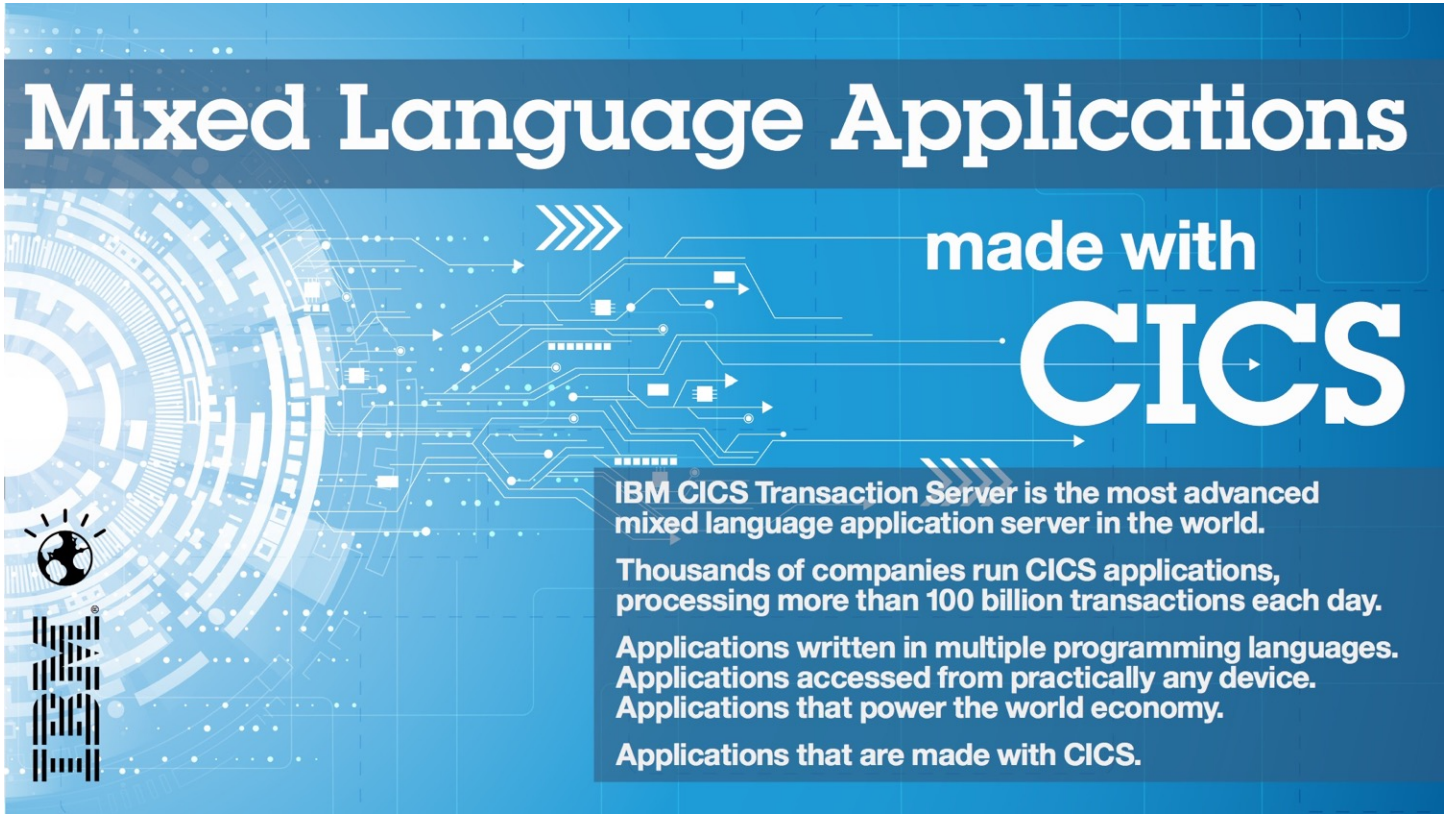
https://ibm.biz/ibmztrial

Generation z

IBM CICS Transaction Server V5.4

ibm.biz/ibmgenz




Mixed Language Applications
made with
CICS
IBM CICS Transaction Server is the most advanced mixed language application server in the world.
Thousands of companies run CICS applications, processing more than 100 billion transactions each day.
Applications written in multiple programming languages.
Applications accessed from practically any device.
Applications that power the world economy.
Applications that are made with CICS.

And that concludes our whistle stop tour of CICS TS

CICS is a high performance mixed language application server

You all now know what a transaction is!

CICS enforces transactionality, and security on everything that runs in it

CICS has been around for near 50 years

We are constantly adding support for cutting edge technologies

Any Questions?

We want your feedback!

- Please submit your feedback online at ….
  - http://conferences.gse.org.uk/2018/feedback/ai
- Paper feedback forms are also available from the Chair person
- This session is AI