2012
IBM System z Technical University

Enabling the infrastructure for smarter computing

CICS TS : Introduction to Applications as first class entities

zAI08

Matthew Webster
CICS meets the cloud
matthew_webster@uk.ibm.com

CICS Cloud Enablement

Monday

0830 *General Session*

Tuesday

1030 **CICS TS : Introduction to Applications as first class entities**

1300 CICS TS: Introduction to Platforms as deployment targets

Wednesday

0900 CICS TS: Application Management Update

1030 CICS TS: Platform Management Update

1300 *CICS in the Cloud: Hands-On Lab*

1615 *CICS Tools Family Update*

Thursday

Pain Points

- *Deploying and un-deploying applications is a high skill complex job due to the number of separate artifacts*
- *Customers would like to see usage / charging, availability / SLA at the application level*
- *Elastic scale is a requirement, but it needs to be managed within the constraints of the customers resources and business environment*

Pain Points

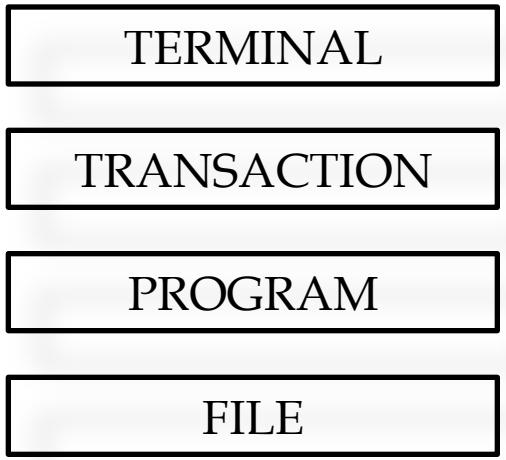
- *Deploying and un-deploying applications is a high skill complex job due to the number of separate artifacts*
- *Customers would like to see usage / charging, availability / SLA's at the application level*
- *Elastic scale is a requirement, but it needs to be managed within the constraints of the customers resources and business environment*

zAI11 CICS TS : Application Management Update

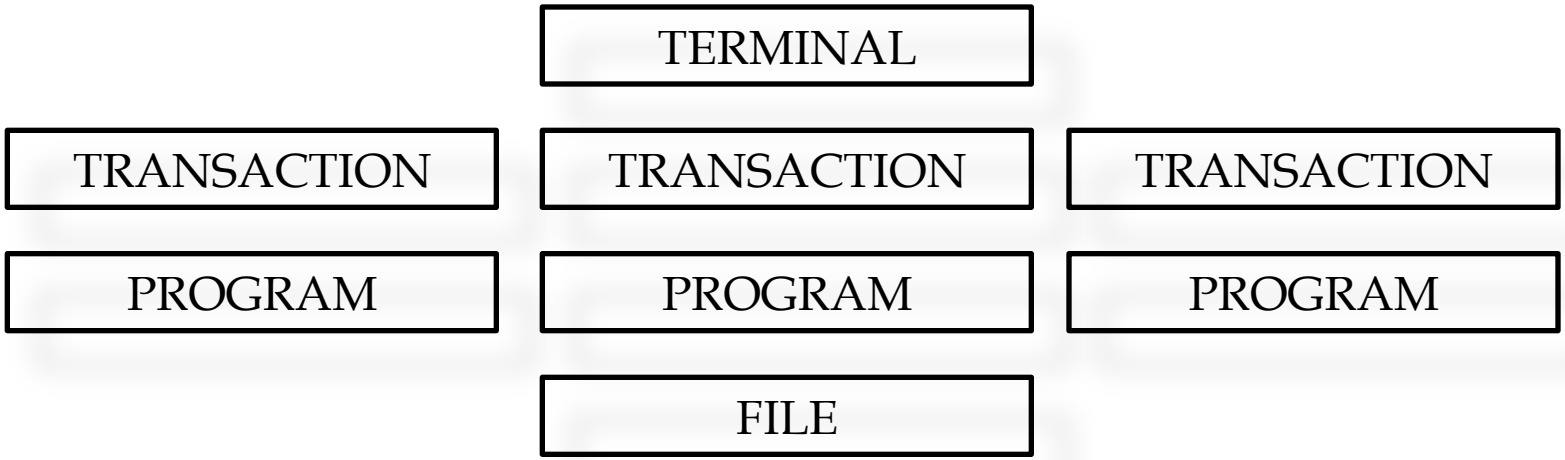
zAI12 CICS TS : Platform Management Update

Once upon a time things were simple

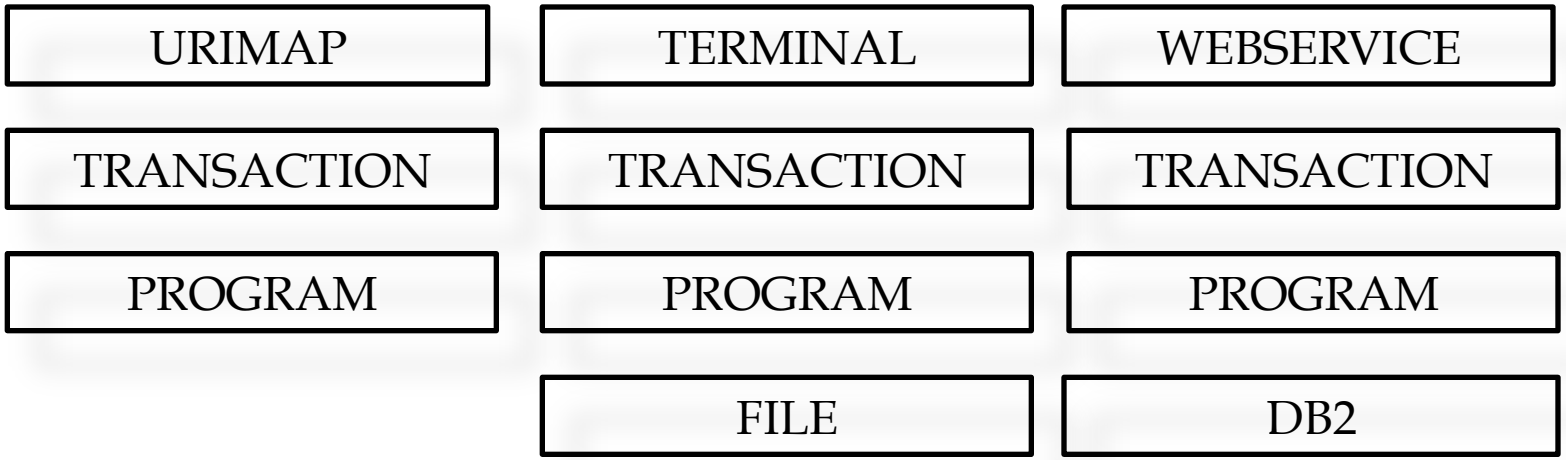
Once upon a time things were simple



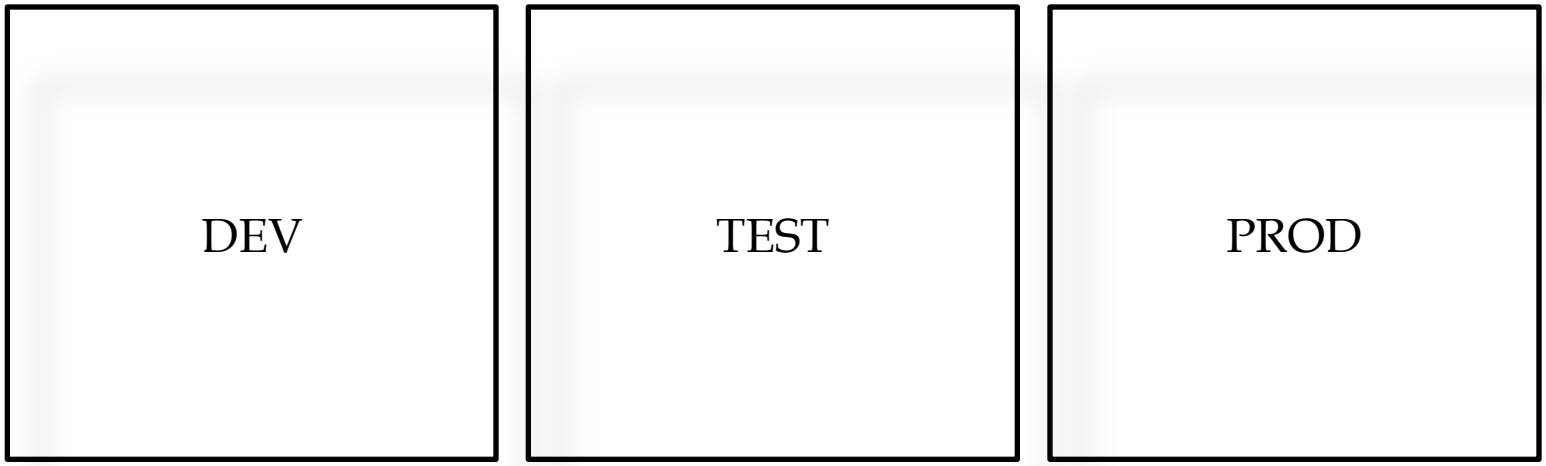
Then capability was added



Then we started getting really clever



And what about those pesky* systems programmers



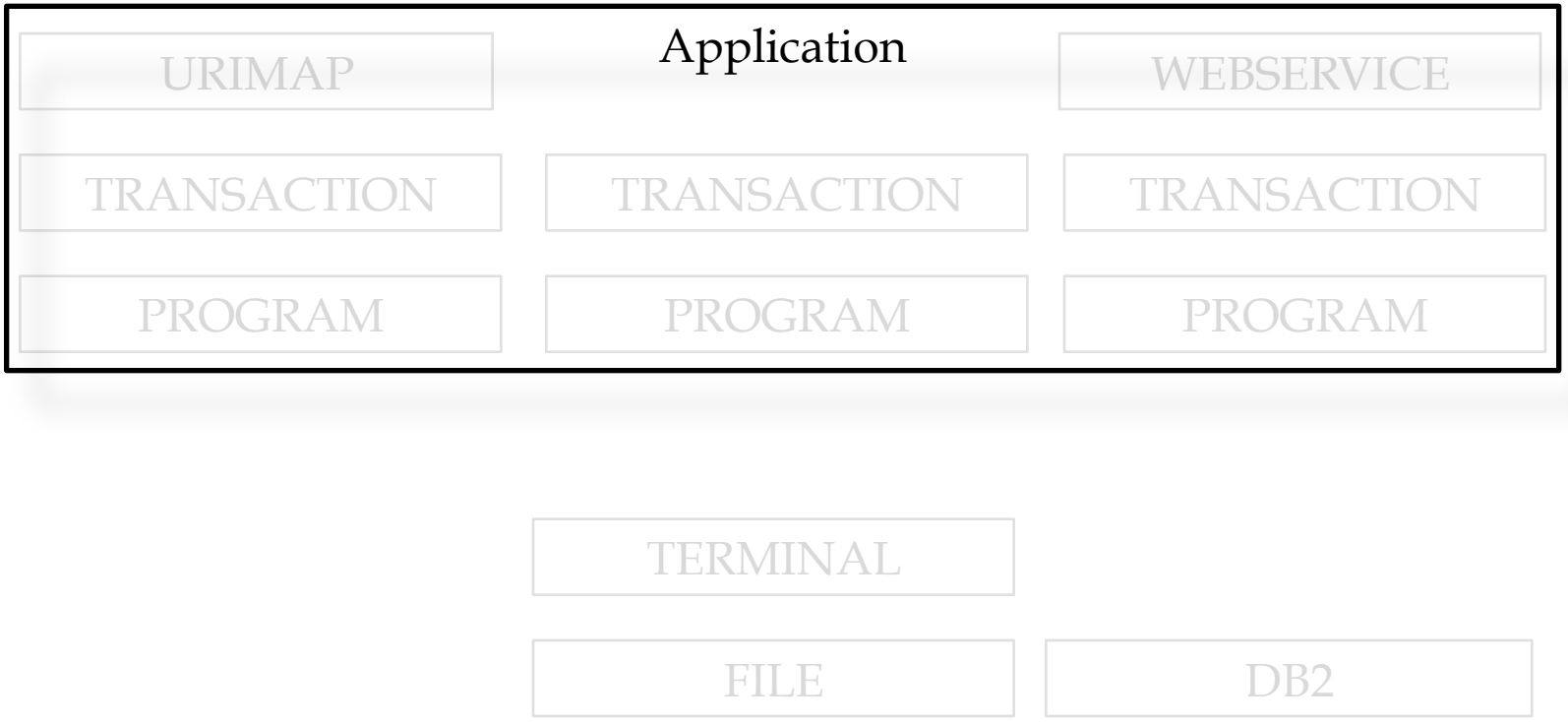
*Troublesome; annoying: *a pesky mosquito*

http://http://www.thefreedictionary.com/pesky

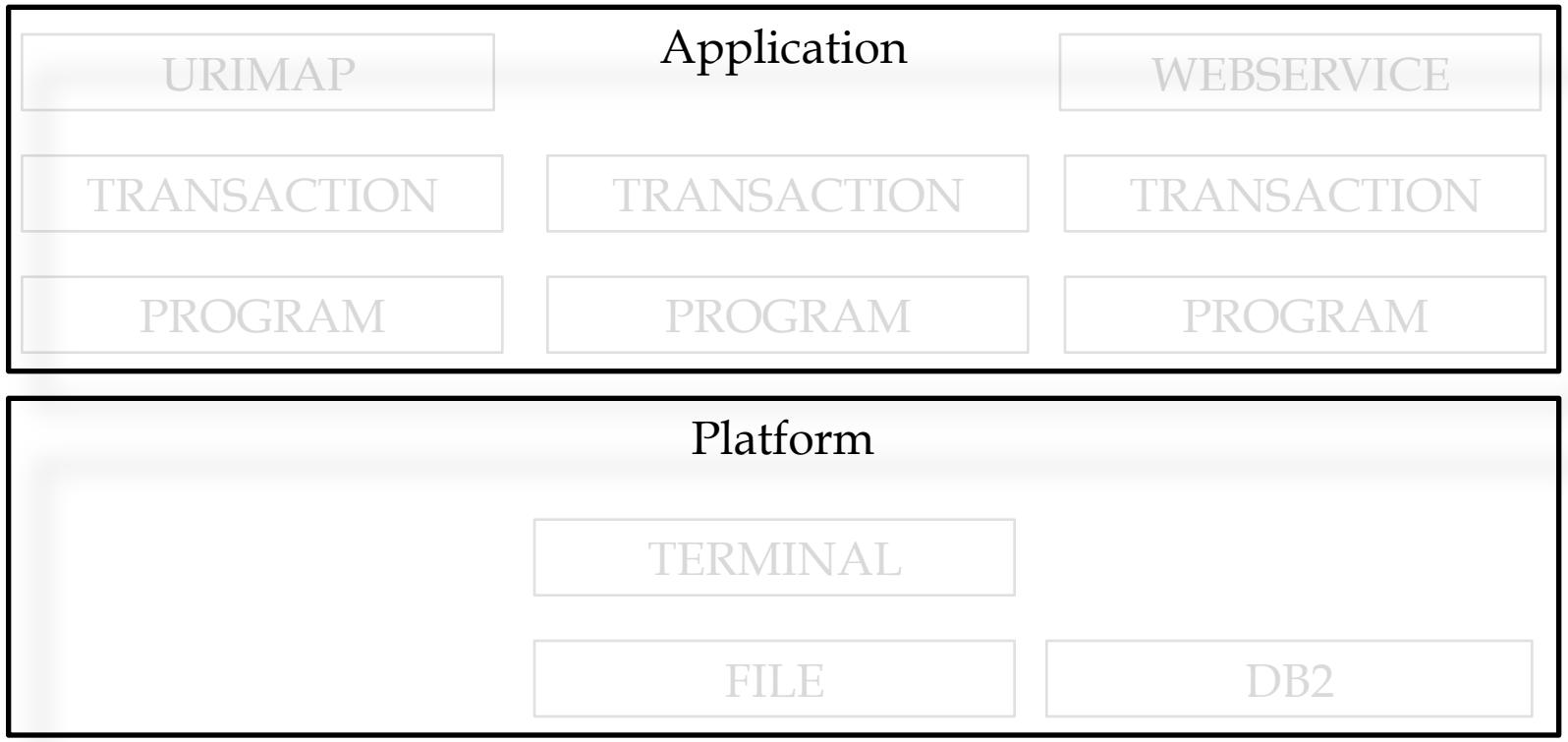
We need things to be simple again

PROGRAM

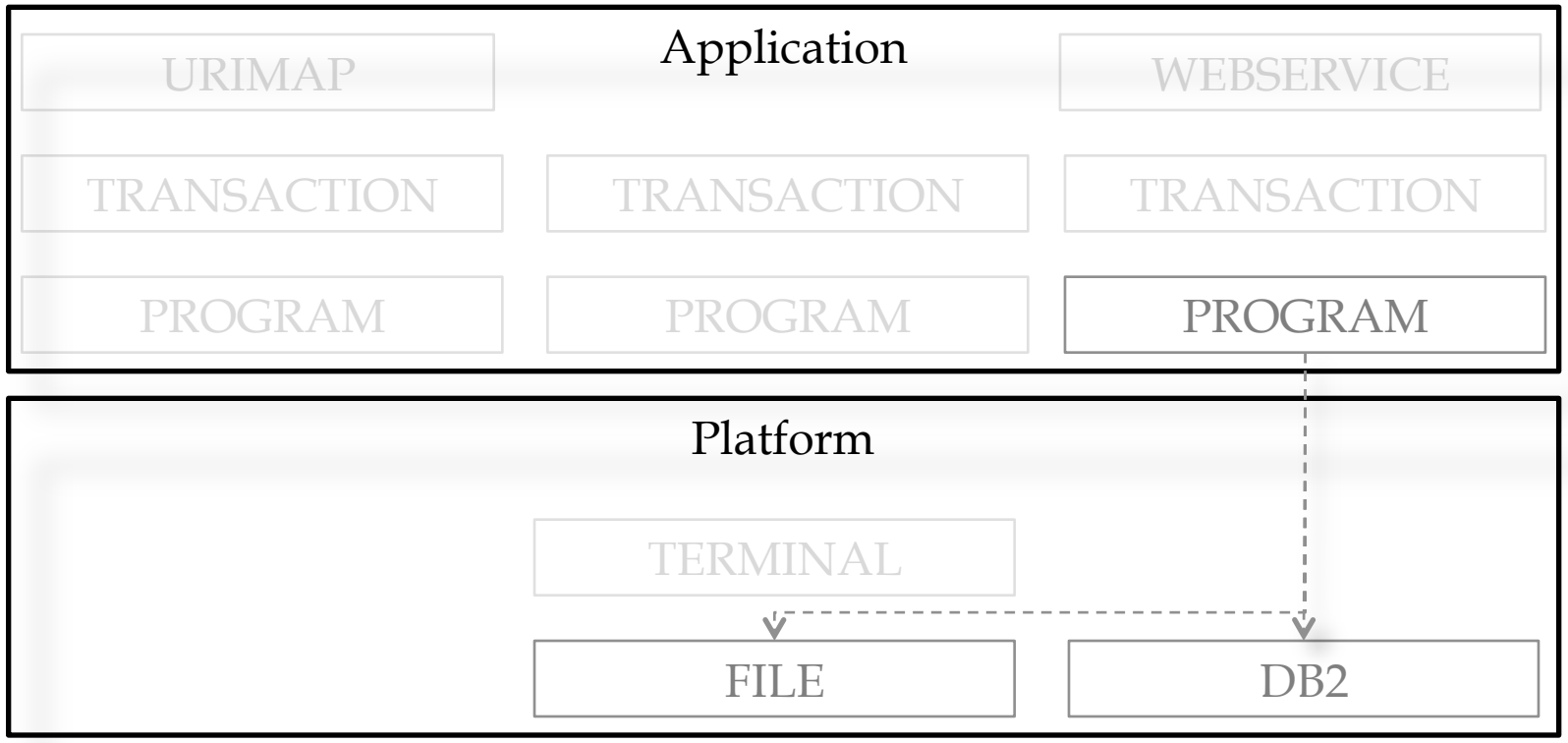
We need things to be simple again: Application



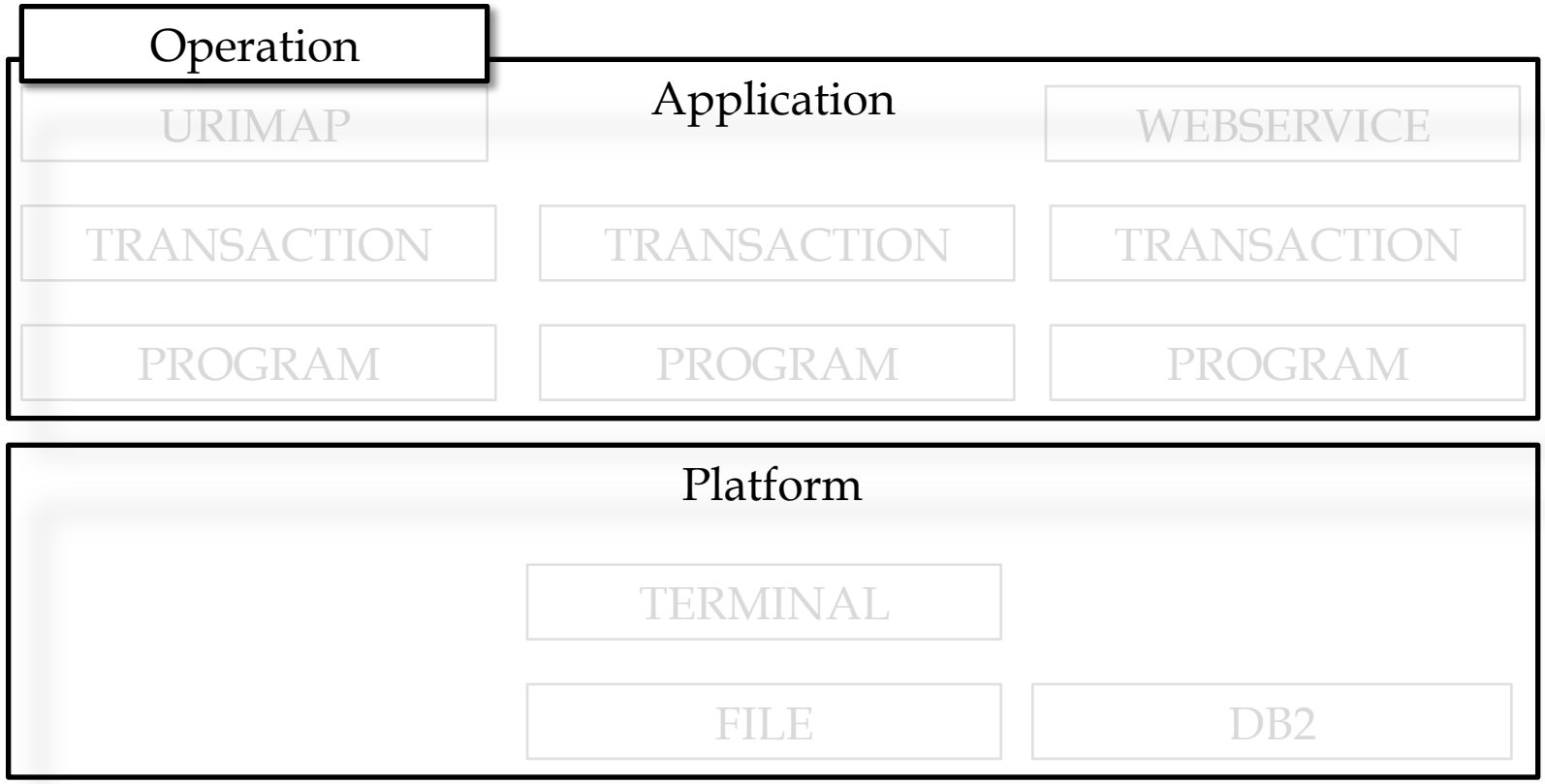
We need things to be simple again: Platform



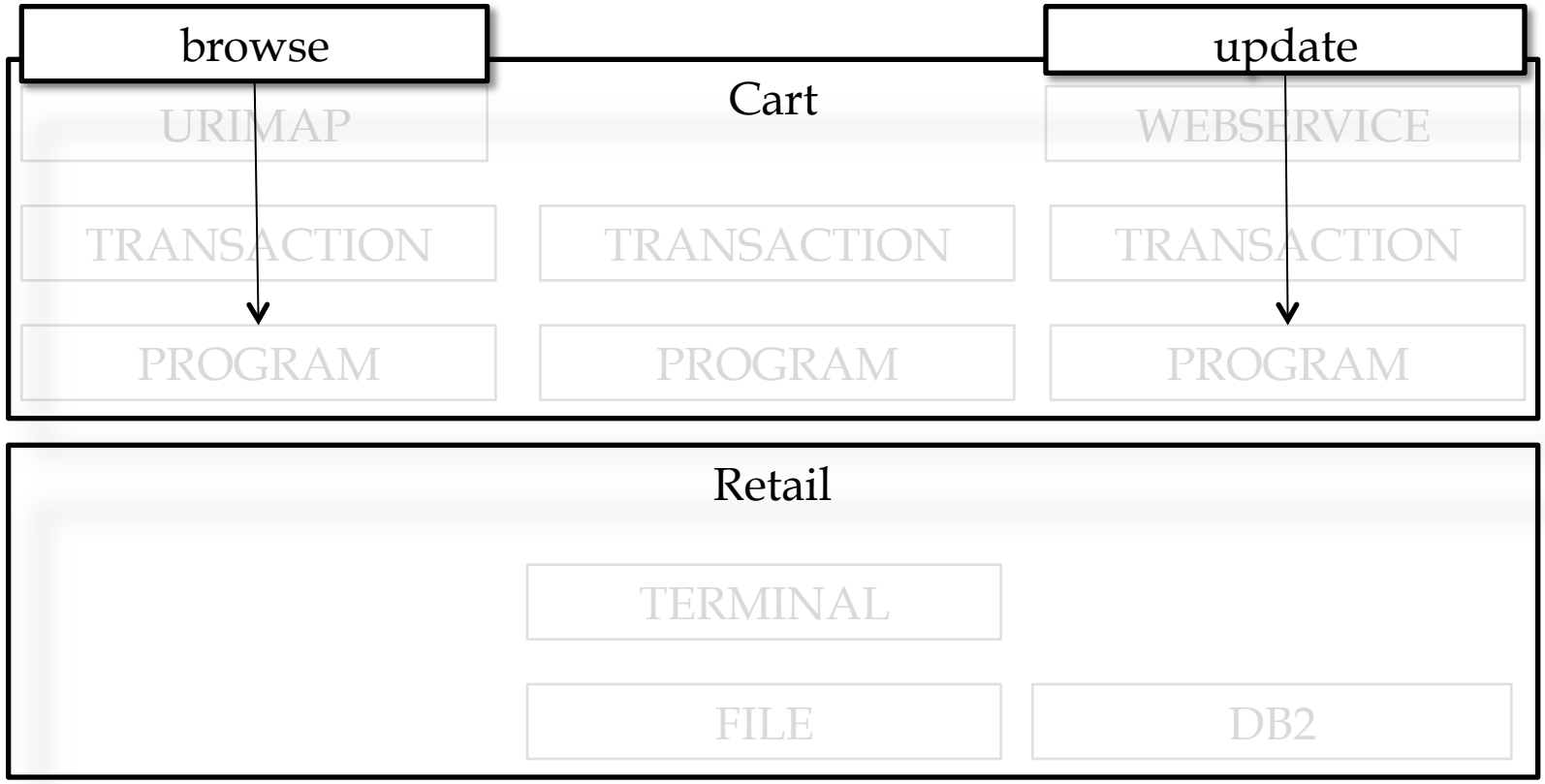
We need things to be simple again: Dependencies



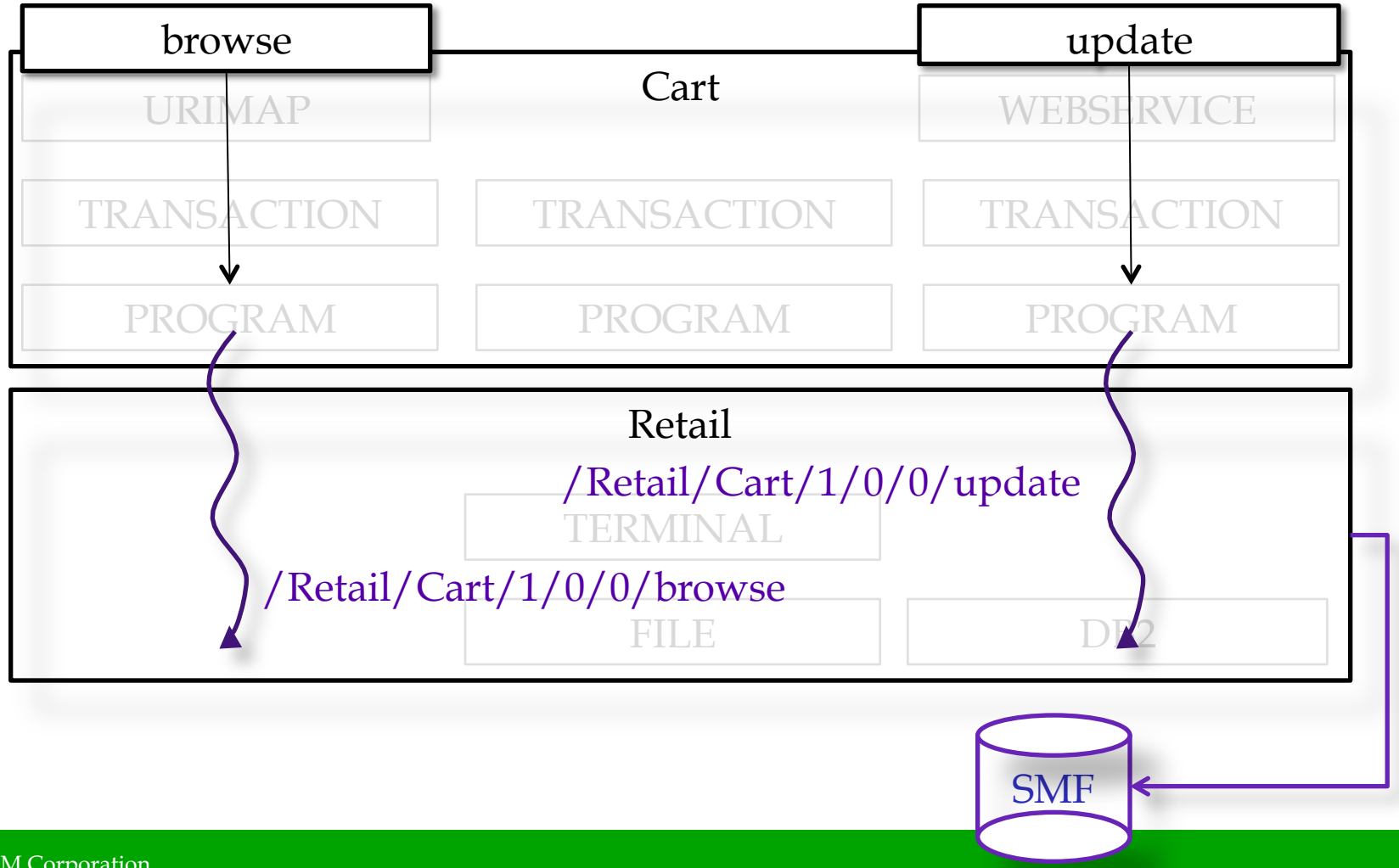
We need things to be simple again: Operations



We need things to be simple again: Entry Points



We need things to be simple again: Application Context



New First Class ~~Concepts~~ Resources

- **Application**
- Platform
- Policy

Application

An Application bundle

A collection of one or more CICS bundles

Life-cycle as a single entity

Measure and control resource usage

Develop in Eclipse/Rational

Share and promote through Source Code Management

Application Package

Name

org.maw.banking.Loans

Version

1.2.1

Resources

LIBRARY, PROGRAM, TRANSACTION, URIMAP
(EVENTBINDING, OSGIBUNDLE, …)

Dependencies

DB2CONNECTION, JVMSERVER, TCPIPSERVICE, …

Entry points

operation: browse, update, …
resource: PROGRAM

Policy

Application Lifecycle

- Create Application bundle project

Application Lifecycle

- Create Application bundle project
- *Create Binding bundle project*

Application Lifecycle

- Create Application bundle project
- *Create Binding bundle project*
- Package CICS bundle(s)

Application Lifecycle

- Create Application bundle project
- *Create Binding bundle project*
- Package CICS bundle(s)
- Export Application package to zFS

Application Lifecycle

- Create Application bundle project
- *Create Binding bundle project*
- Package CICS bundle(s)
- Export Application package to zFS

- INSTALL Application onto a Platform
- ENABLE/DISABLE Application
- DISCARD Application

Application Lifecycle

- Create Application bundle project
- *Create Binding bundle project*
- Package CICS bundle(s)
- Export Application package to zFS

- INSTALL Application onto a Platform
- ENABLE/DISABLE Application
- DISCARD Application

- Application status (DISABLED | ENABLING | …)

Application Context

- Manage Application
- Measure & control resource usage
- Associate Task with Application operation
  - PROGRAM
- Flow from Task to Task & Region to Region
  - MRO, IPIC
- Recorded in monitoring data
  - Platform, Application, Version (major.minor.micro), Operation

Version

Semantic versioning
- major: backward incompatible change
- minor: backward compatible change
- micro: bug fix

Resources
- Application
- CICS bundle
- *OSGi bundle*

Life-cycle
- Development
- Deployment
- Operations
- Planning

http://www.osgi.org/wiki/uploads/Links/SemanticVersioning.pdf

CICS Application vs. CPSM BAS

Simplified abstraction

Offline representation

Separation of DevOps* roles

Complete lifecycle

Versions

Security

Management, measurement and control

* "… communication, collaboration and integration between software developers and information technology(IT) professionals …"

http://en.wikipedia.org/wiki/DevOps

Application Questions

Can I define single application package that moves from development, through test to production unchanged?

Yes

Can I tell what's changed between two versions of the same application at any point in the life-cycle?

Yes

Can I use source code management (SCM) to manage application lifecycle?

Yes

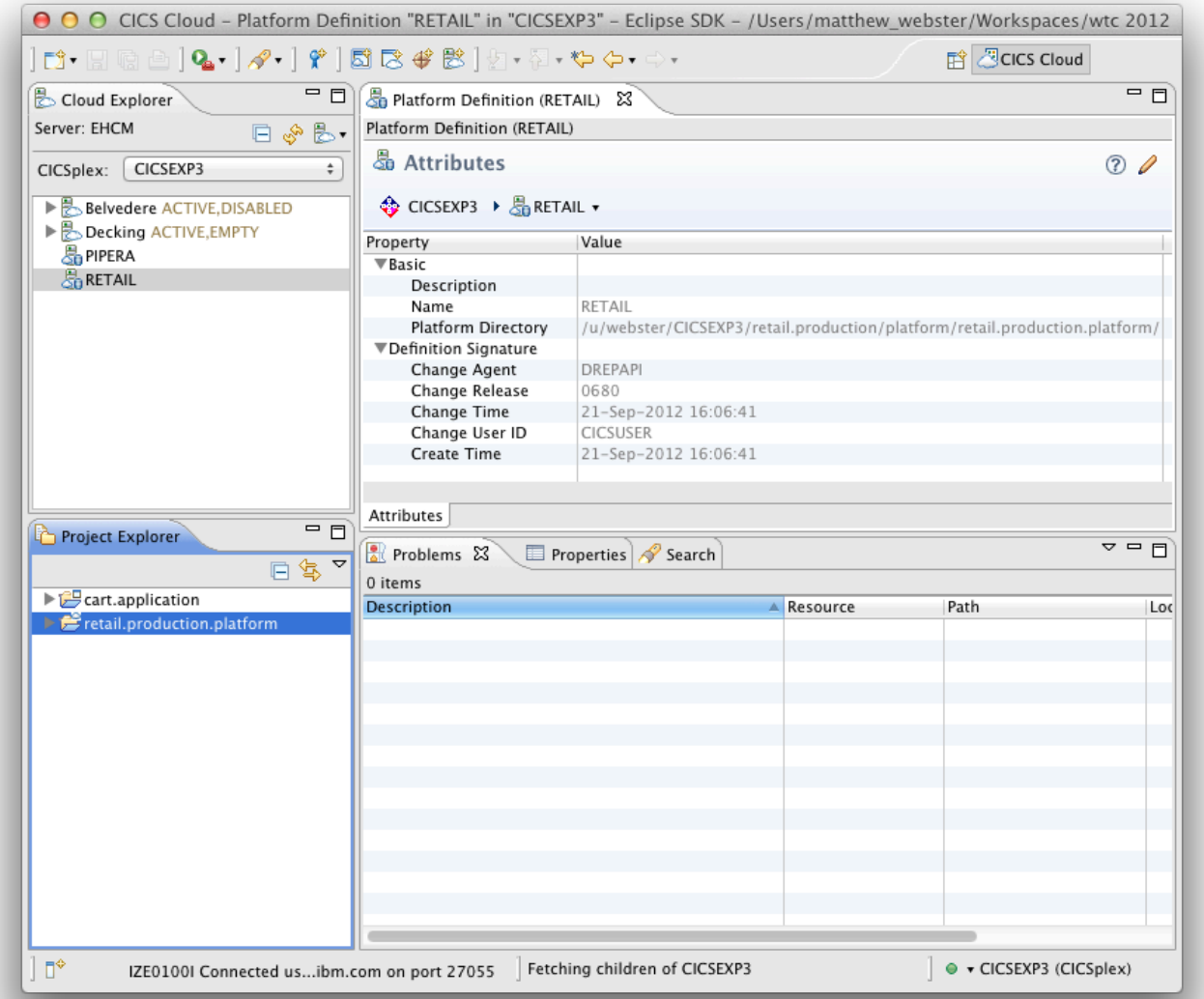
DevOps Roles

- **Application Developer**
- System Programmer
- *Application Deployer*

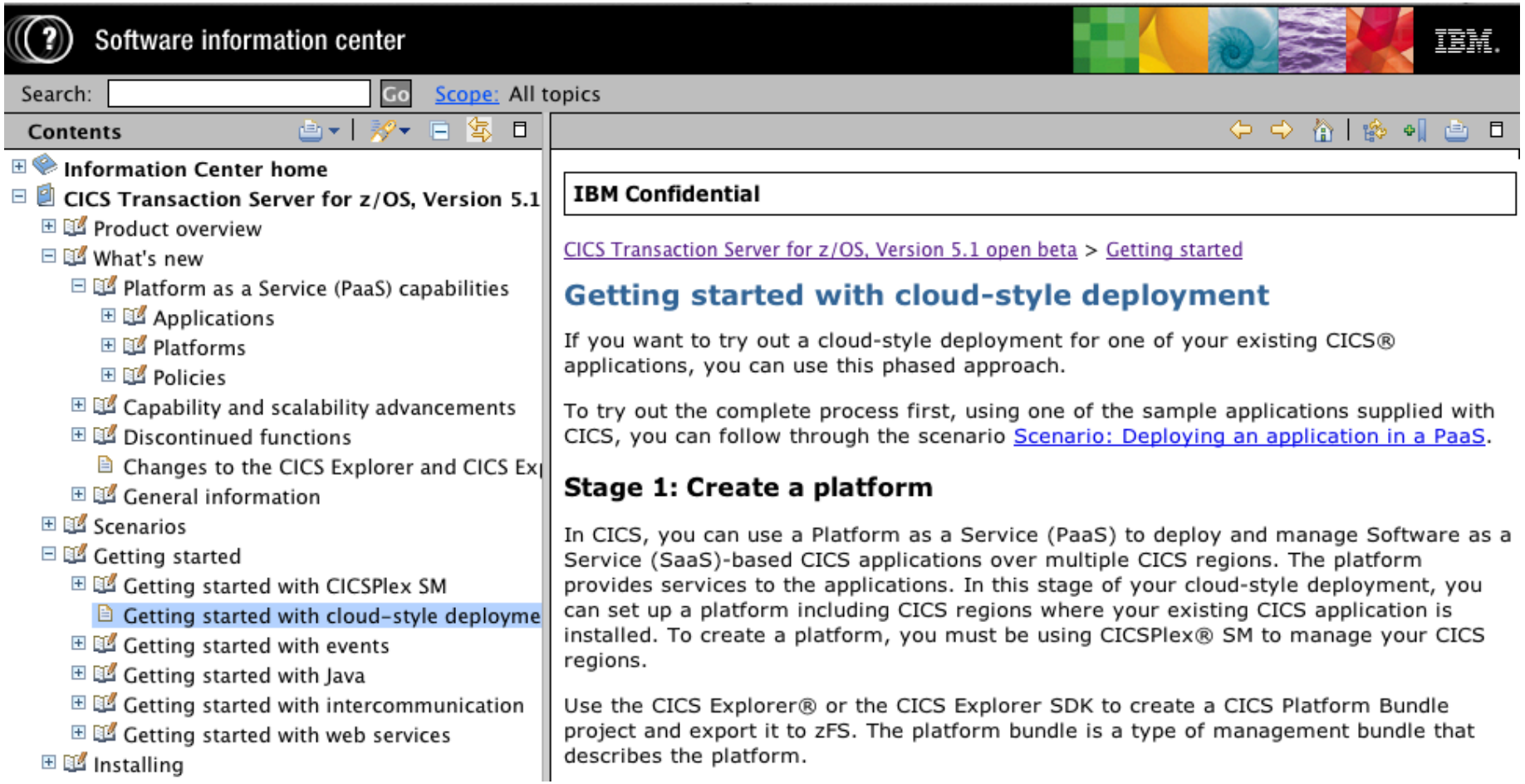
DevOps Roles

- **Application Developer**
  - **Resources**
  - **Dependencies**
  - **Entry points**
  - **Policy**
- System Programmer
  - Topology
  - Services
  - Policy
- *Application Deployer*
  - *Deployment*
  - *Resources*
  - *Policy*

Demonstration



Information Center



Getting started with cloud-style deployment

- **Stage 1:** Create a platform
- **Stage 2:** Create an application
- **Stage 3:** Add application entry points
- **Stage 4:** Add resources for the application
- **Stage 5:** Add a policy

Getting started with cloud-style deployment

- **Stage 1:** Create a platform
- **Stage 2: Create an application**
- **Stage 3: Add application entry points**
- **Stage 4: Add resources for the application**
- **Stage 5:** Add a policy

Stage 2: Create an application

Single description of an Application: capabilities, composition and dependencies

Shared concept between all user roles (stakeholders): developer, system programmer, end user

Simplify management of resource and service status

Stage 3: Add application entry points

Declare end-user value

Measure Application resource consumption

Prepare Application for Policy

Stage 4: Add resources for the application

Use Application artifacts in a workflow to manage the development lifecycle

Simplify resource deployment and undeployment

Allow Application and Platform to evolve independently (DevOps)

Summary

New Application resource

Simplified development and deployment lifecycle

Simplified management of runtime status

Questions?

More Information

- "CICS TS"
  - http://www.ibm.com/cics/
- developerWorks Community
  - https://www.ibm.com/developerworks/mydeveloperworks/blogs/cicsdev/
- "Did you say mainframe?! " Podcasts
  - http://itunes.apple.com/us/podcast/did-you-say-mainframe-!/id275831334

CICS TS Highlights

Monday	
1030	CICS Portfolio update
1300	CICS TS: A Technical Overview
Tuesday	
0900	CICS TS: Touching the Cloud – Introducing Enterprise Services
1030	CICS TS: Introduction to Applications as first class entities
1030	*CICS Portfolio update*
1300	**CICS TS: Introduction to Platforms as deployment targets**
1430	*CICS TS: A Technical Overview*
Wednesday	
0900	CICS TS: Application Management Update
1030	CICS TS: Platform Management Update
1300	CICS in the Cloud: Hands-On Lab
1615	CICS Tools Family Update
Thursday	
1030	CICS TS: Support for WebSphere Application Server Liberty profile
1300	CICS TS: Java and the JVM Server
1430	CICS TS: Scalability Enhancements