Counting on the Cloud:
Cloud Native Transactions

Ian Robinson, DE & CTO IBM App Platform

IBM **Cloud**

Please note

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

As Little as 10% of application code is for the happy path.

Designing for handling errors is hard

It is expensive to test
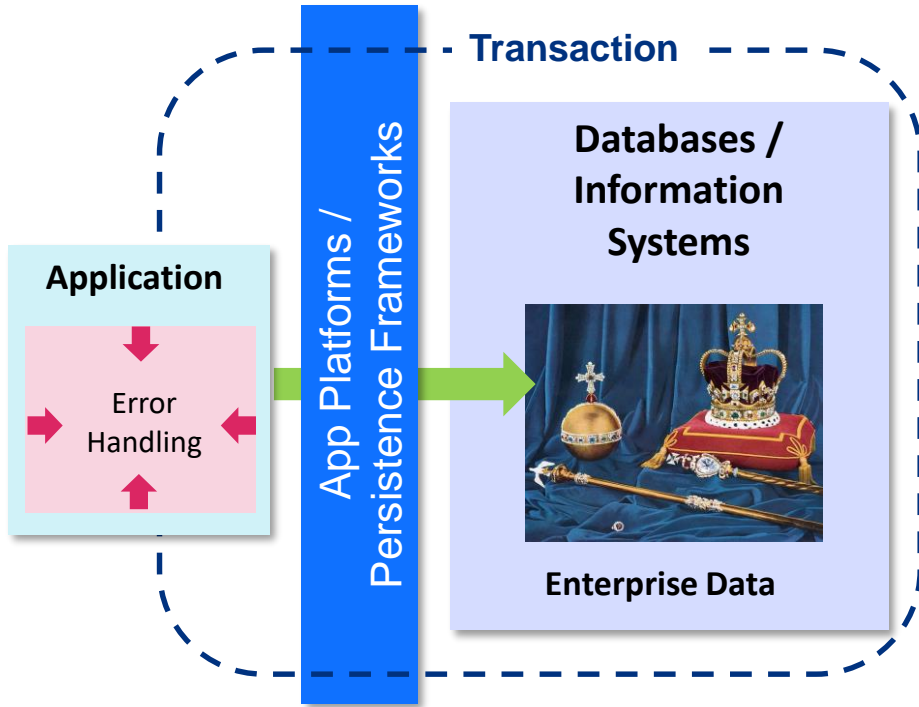
It's what most post-release investment is spent on

McConnell, 2004: http://aroma.vn/web/wp-content/uploads/2016/11/code-complete-2nd-edition-v413hav.pdf

Transactions simplify how applications interact with data

Through access patterns that provide guarantees about the data to the application

Most of the time, we just don't notice transactions.

Because we take them for granted
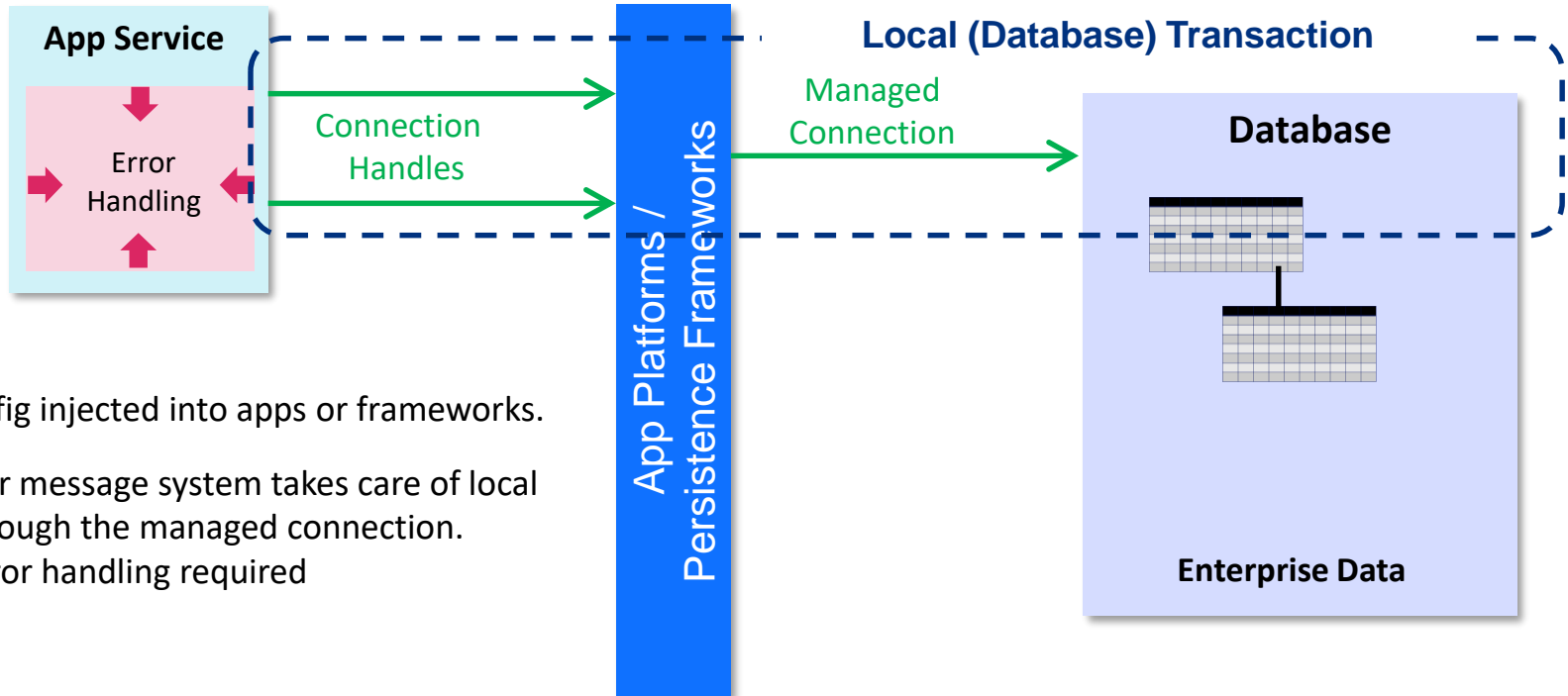
But we notice them when they aren't there.

Every step on the digital evolution of the last 50 years has introduced:

- **New opportunity** for application innovation through capability that wasn't available before

- **New complexity** in error scenarios to be handled by applications as a consequence

- **New strategies** for applications to use to eliminate or simplify that complexity

Local Transactions Used To Be Hard

but then became second nature

App Service

Error Handling

App Platforms / Persistence Frameworks

Connection Handles

Managed Connection

Local (Database) Transaction

Database

Enterprise Data

Connection config injected into apps or frameworks.

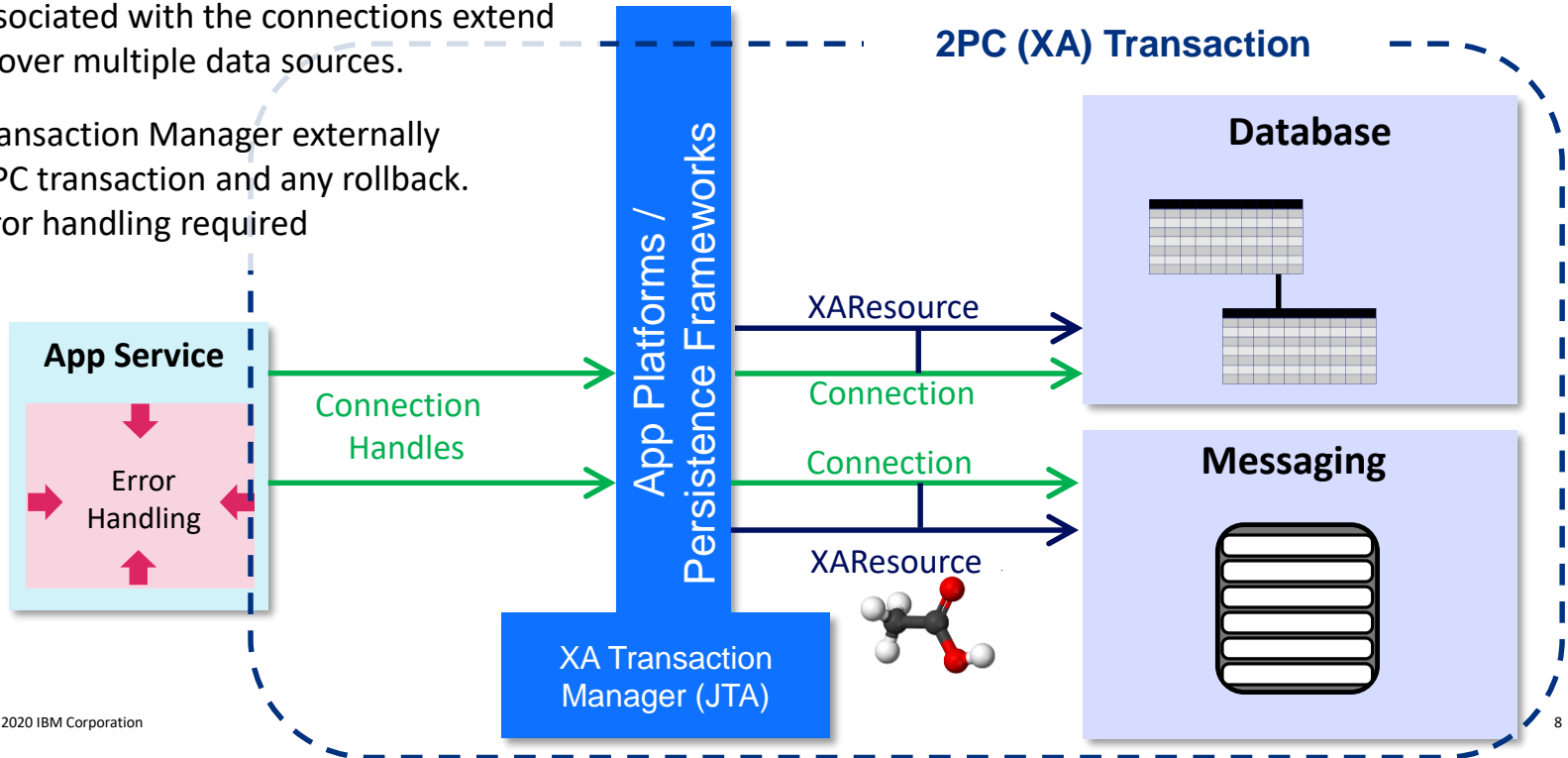Back end data or message system takes care of local transactions through the managed connection.
Minimal app error handling required

XA Transactions Used To Be Hard

but then became easy

XAResources associated with the connections extend the transaction over multiple data sources.

App Platform Transaction Manager externally coordinates a 2PC transaction and any rollback. Minimal app error handling required
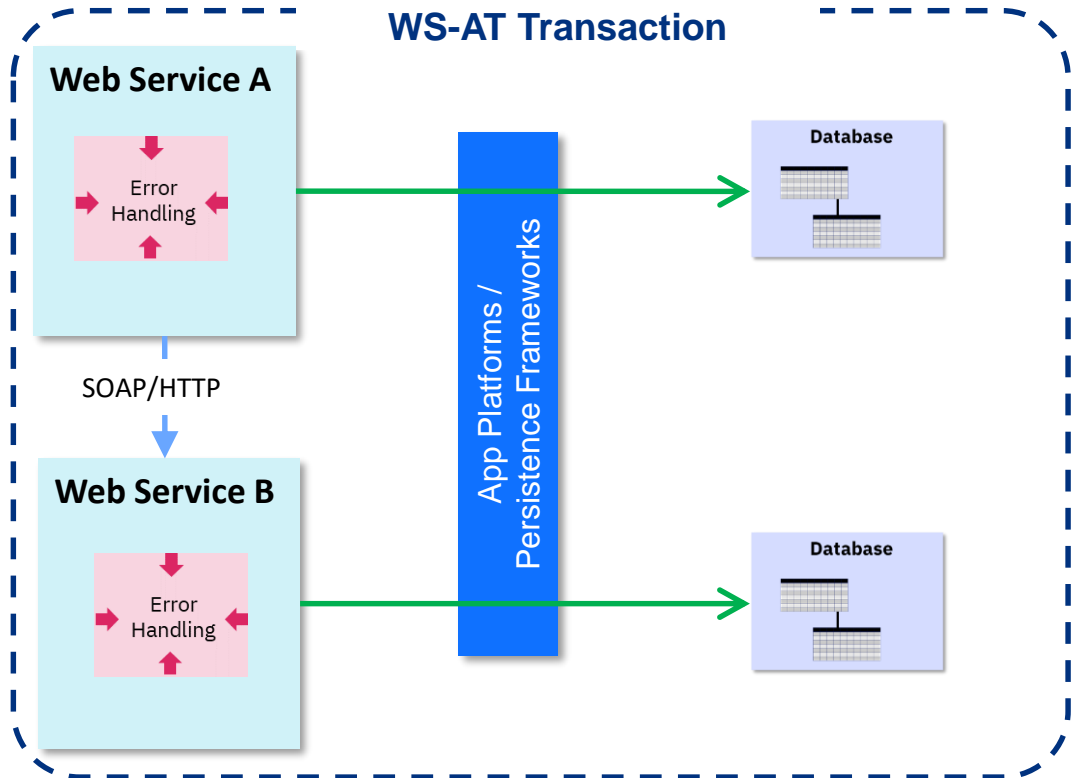
2PC (XA) Transaction

App Platforms / Persistence Frameworks

App Service

Error Handling

Connection Handles

XA Transaction Manager (JTA)

XAResource

Connection

Connection

XAResource

Database

Messaging

SOA and Web Services Used To Be Hard

but then became easier

Web Service transactions extended the reach of 2PC transactions across multiple applications.

WS-AT still kept application error handling minimal.

Tight-coupling of the services limited use to fine-grained SOA.
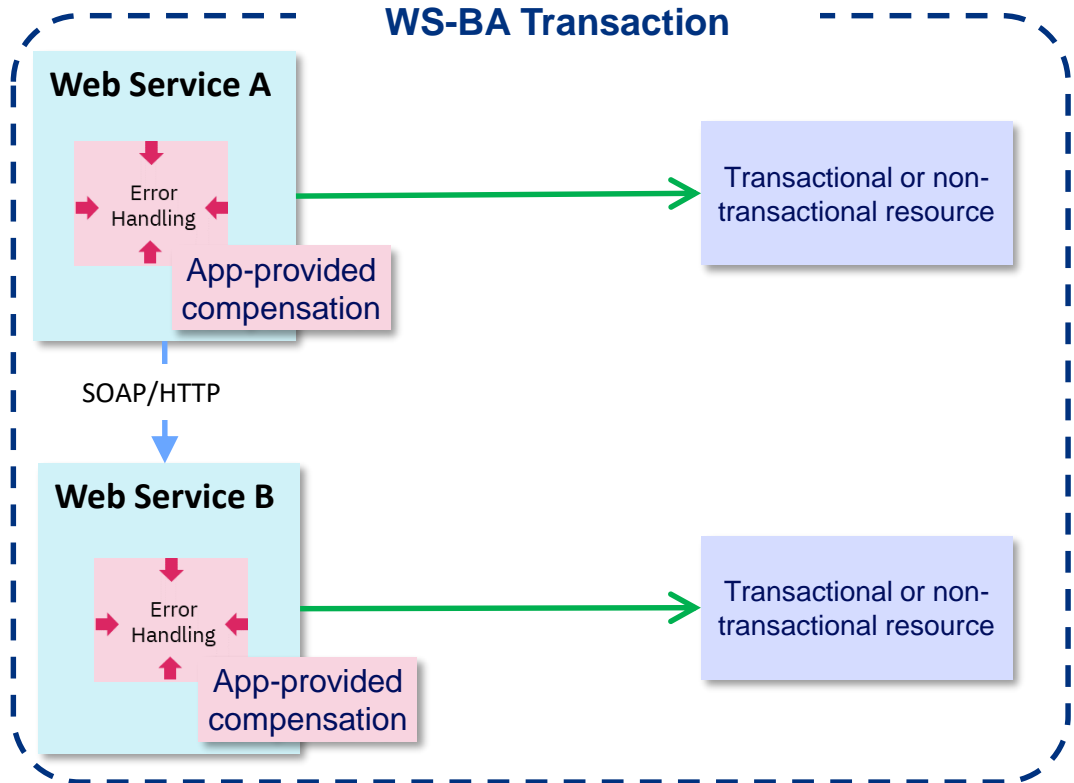

© 2020 IBM Corporation

SOA and Web Services Used To Be Hard

and increased the use of more loosely coupled transactions

WS-Business Activity made the use of compensating transactions more common.

Part of WebSphere since V6.1 (2006).

Application logic becomes part of the failure-handling.

Simple model for registering application compensation handlers. The platform ensures these are driven following any failure.

WS-BA Transaction

Web Service A

Error Handling

App-provided compensation

SOAP/HTTP

Web Service B

Error Handling

App-provided compensation

Transactional or non-transactional resource
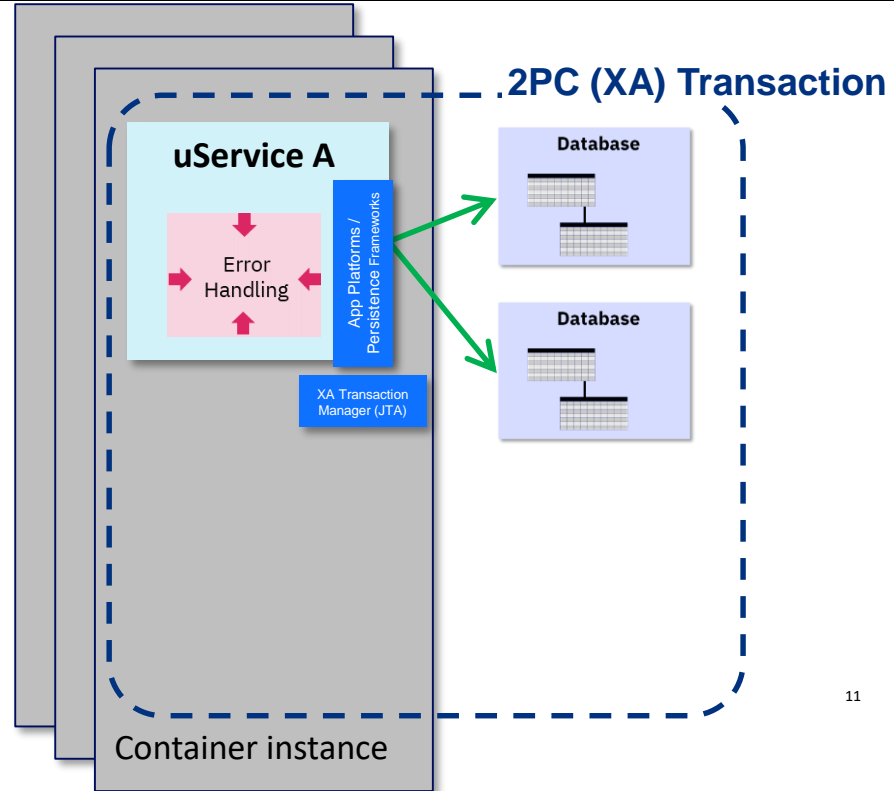
Transactional or non-transactional resource

Transactions in Cloud Containers

Introduce new challenges

2PC transactions do work in containers when transactional recovery logs are properly configured in persistent storage.

Transaction peer recovery takes care of scale-down scenarios. Beware of scaling to zero as this may leave resources locked until there is a new container instance.

But what if the two DBs here were accessed by difference microservices?

2PC (XA) Transaction

uService A

Error Handling

App Platforms / Persistence Frameworks

XA Transaction Manager (JTA)

Database

Database

Container instance

11

Transactions in Cloud Containers

Introduce new challenges

If app design and error handling assumes 2PC, the two services are tightly coupled. There really are not 2 separate microservices here as these have a single lifecycle and are not independently scalable. You would never do what is shown here.
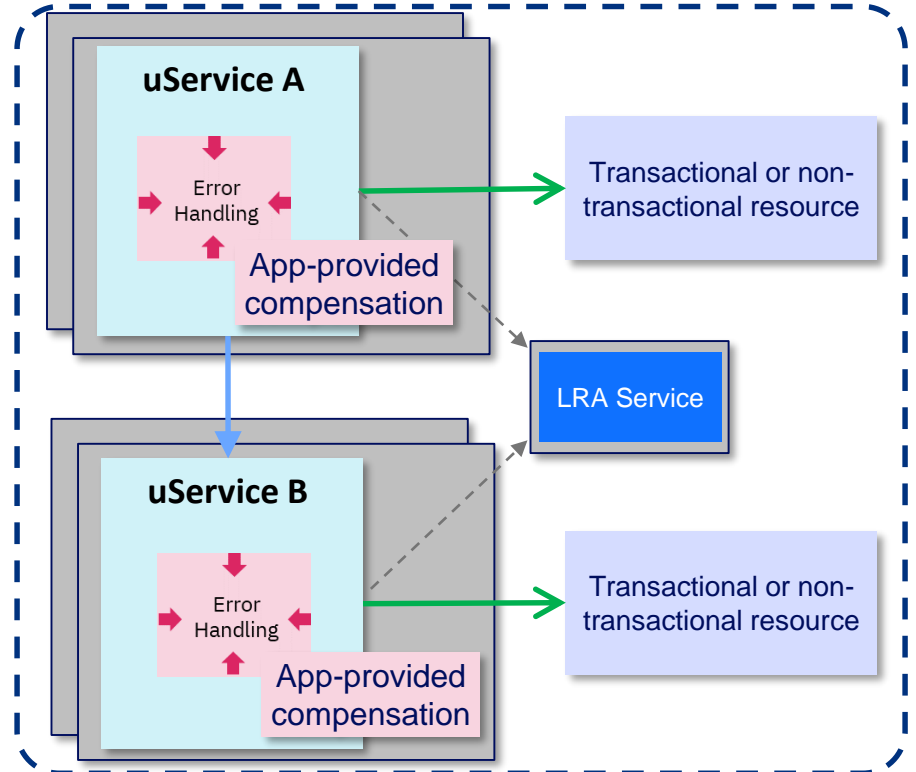
A different transaction model is required.
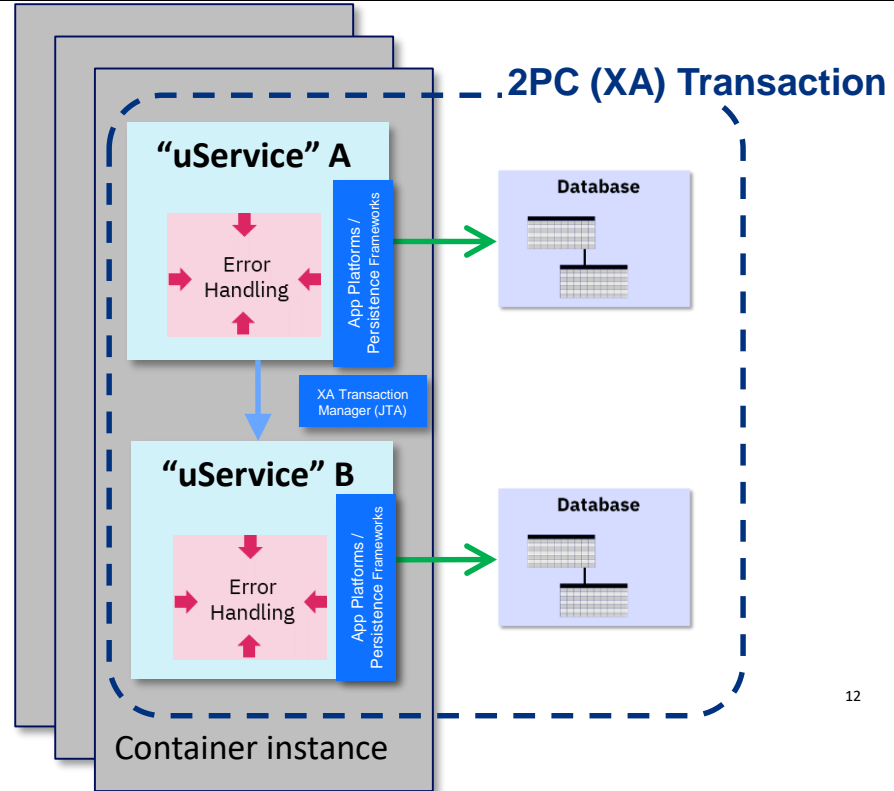
Transactions in Cloud Containers

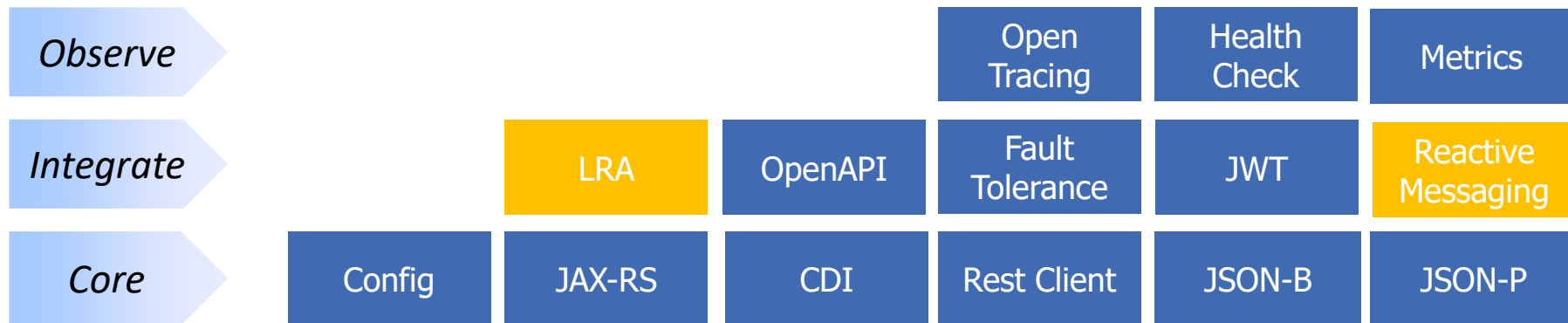Eventual Consistency and Long-Running Actions

MicroProfile **Long-Running Actions (LRA)** is a model for achieving eventual consistency of data across microservices.

- Derived from the WS-BA model for Web Service Business Activities

- Simple annotation-based model.

uService A

Error Handling
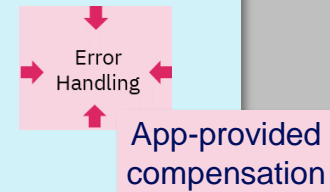
App-provided compensation

Transactional or non-transactional resource

LRA Service

uService B

Error Handling

App-provided compensation

Transactional or non-transactional resource

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

https://microprofile.io/

Observe				Open Tracing	Health Check	Metrics
Integrate		LRA	OpenAPI	Fault Tolerance	JWT	Reactive Messaging
Core	Config	JAX-RS	CDI	Rest Client	JSON-B	JSON-P

LRA Application Annotations

uService A

```
@Path("/")
@ApplicationScoped
public class TravelBookingService
{
    @LRA(LRA.Type.REQUIRES_NEW)
    @Path("/bookflight")
    @POST
    public Response bookFlight(@HeaderParam(…) URI LRAid)
    {…}

    @Complete
    @Path("/complete")
    @PUT
    public Response completeWork(@HeaderParam(…) URI LRAid)
    {…}

    @Compensate
    @Path("/compensate")
    @PUT
    public Response compensateWork(@HeaderParam(…) URI LRAid)
    {…}
}
```
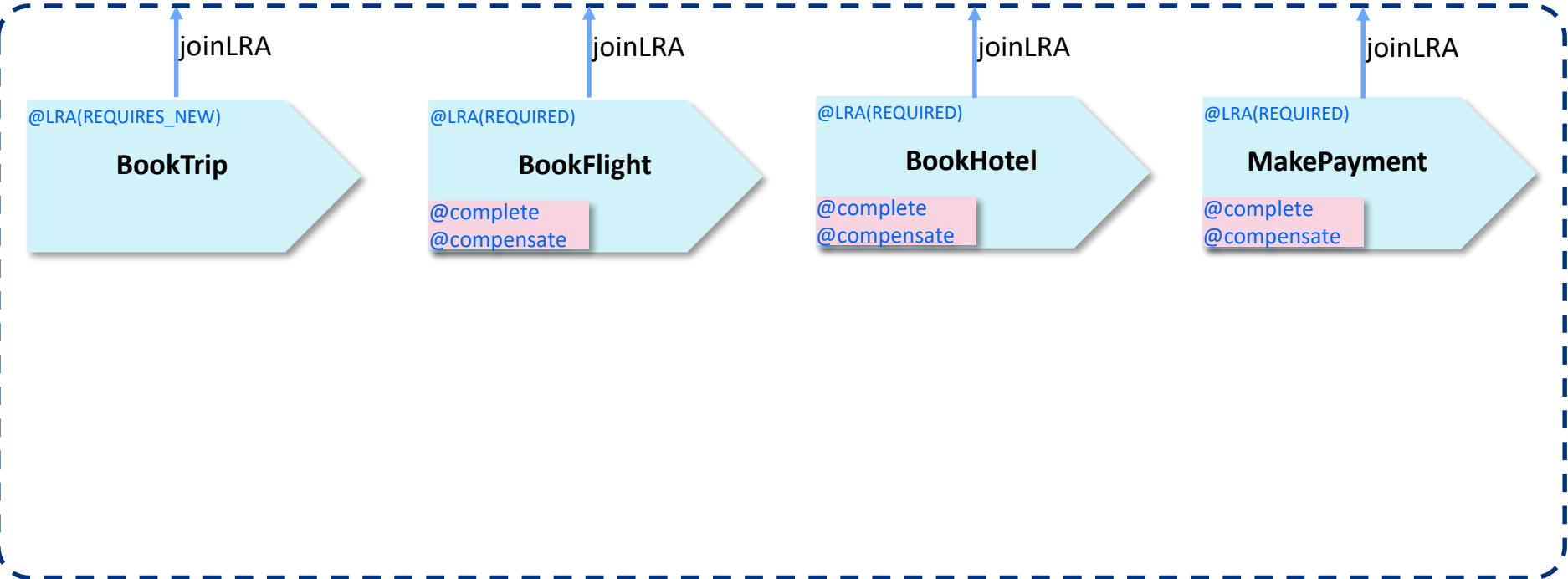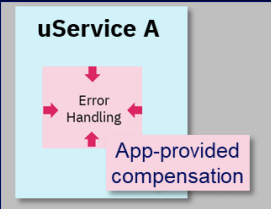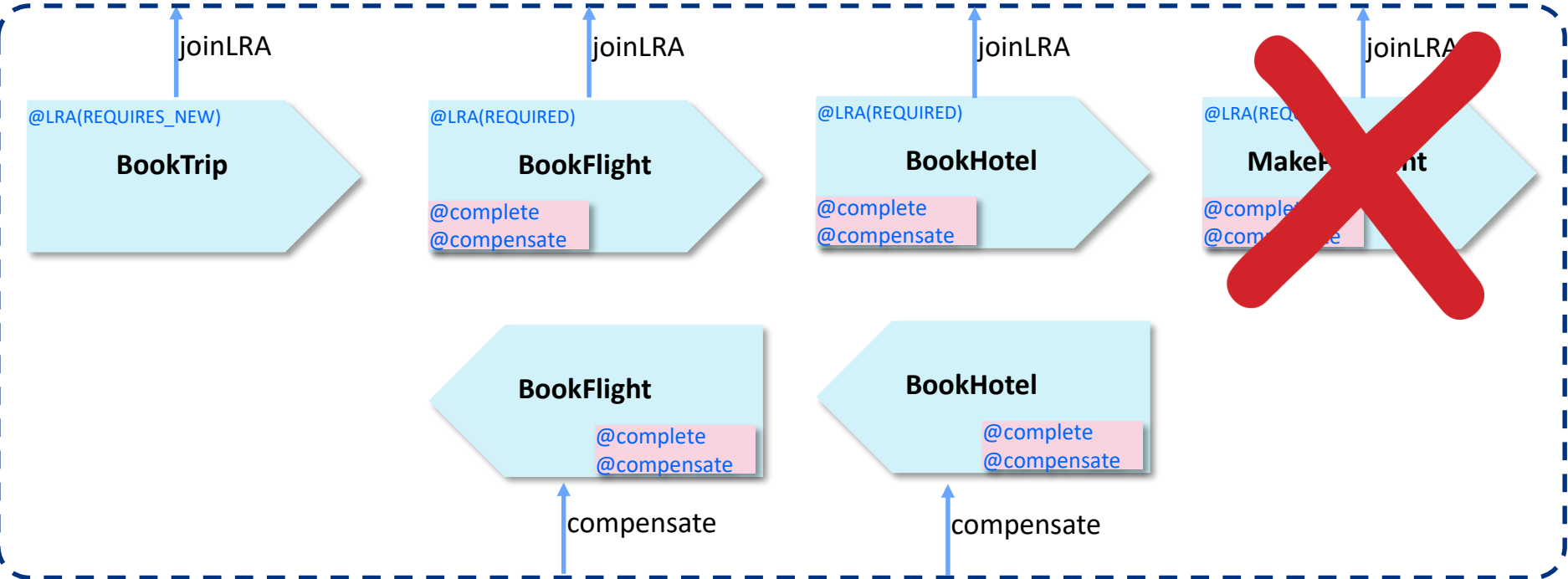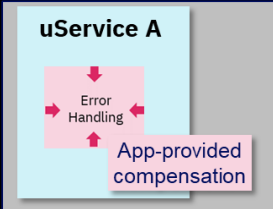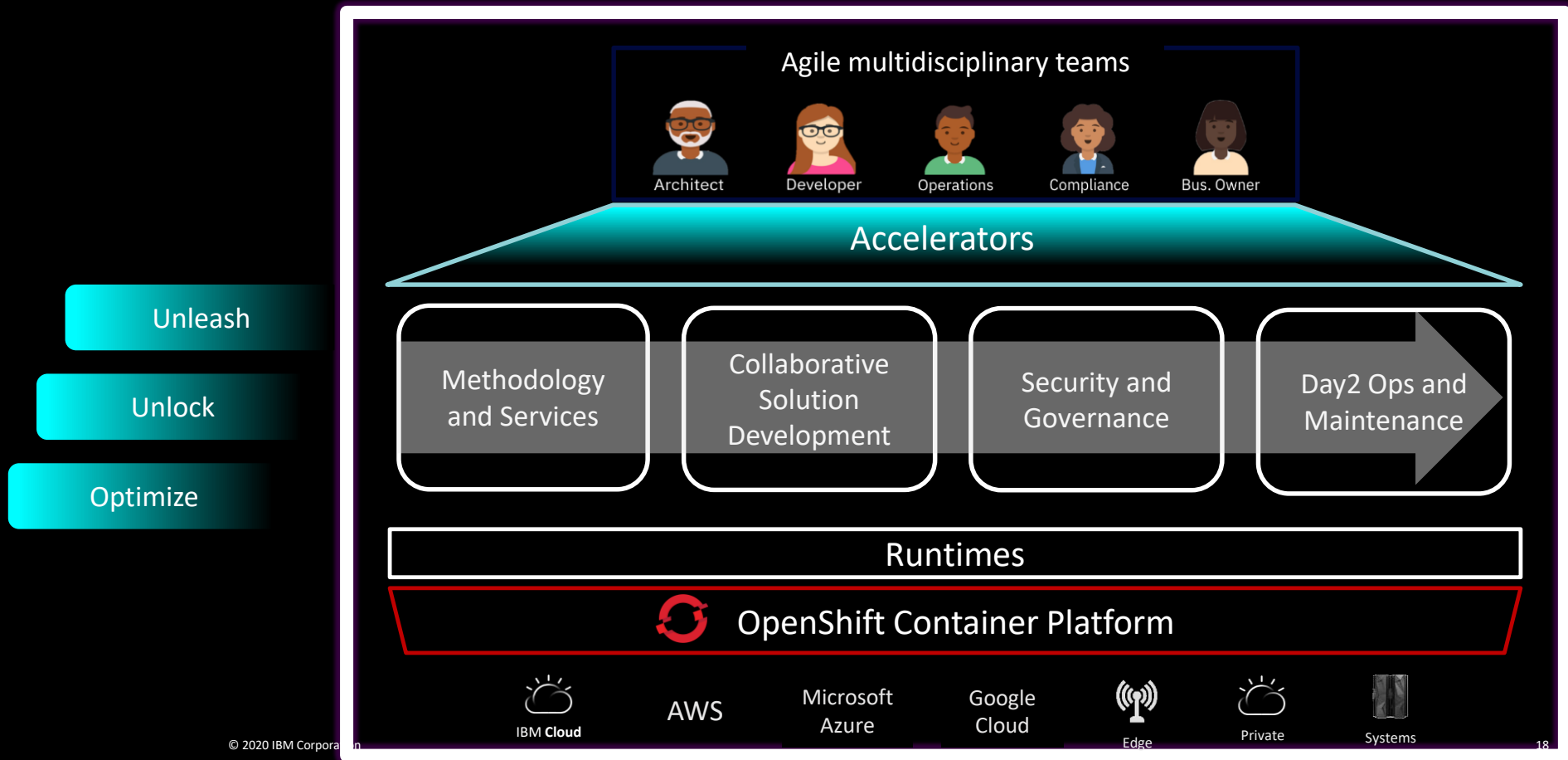
uService A

Error Handling

App-provided compensation

Sagas Using MicroProfile LRA

uService A
Error Handling
App-provided compensation

joinLRA

joinLRA

joinLRA

joinLRA

@LRA(REQUIRES_NEW)
BookTrip

@LRA(REQUIRED)
BookFlight
@complete
@compensate

@LRA(REQUIRED)
BookHotel
@complete
@compensate

@LRA(REQUIRED)
MakePayment
@complete
@compensate

Sagas Using MicroProfile LRA

uService A

Error Handling

App-provided compensation

joinLRA

@LRA(REQUIRES_NEW)
BookTrip

joinLRA

@LRA(REQUIRED)
BookFlight
@complete
@compensate

joinLRA

@LRA(REQUIRED)
BookHotel
@complete
@compensate

joinLRA

@LRA(REQ...)
MakeP...nt
@comple...
@compe...

BookFlight
@complete
@compensate

compensate

BookHotel
@complete
@compensate

compensate

Transactions and Accelerators: IBM Cloud Pak for Applications

Agile multidisciplinary teams

Architect | Developer | Operations | Compliance | Bus. Owner

Accelerators

Unleash

Unlock

Optimize

| Methodology and Services | Collaborative Solution Development | Security and Governance | Day2 Ops and Maintenance |

Runtimes

OpenShift Container Platform

IBM Cloud | AWS | Microsoft Azure | Google Cloud | Edge | Private | Systems

MicroService Transactions Used To Be Hard
but we're making them simpler

Transactions Accelerator makes it easy to adopt the LRA pattern

- Transaction Manager deployed as a pre-configured, bindable service

- μServices deployed using pre-configured application stacks and templates

- Metrics, Observability and Telemetry pre-configured

- Resilience and availability pre-configured

- **Application Stack**
- Pre-configured for:
  - Manager use
  - Resilience
  - Observability

uService A

Error Handling

App-provided compensation

Transactional or non-transactional resource

LRA Service

- **Deployable Service**
- Pre-configured for:
  - Resilience
  - Observability

- **Application Stack**
- Pre-configured for:
  - Manager use
  - Resilience
  - Observability

uService B

Error Handling

App-provided compensation

Transactional or non-transactional resource
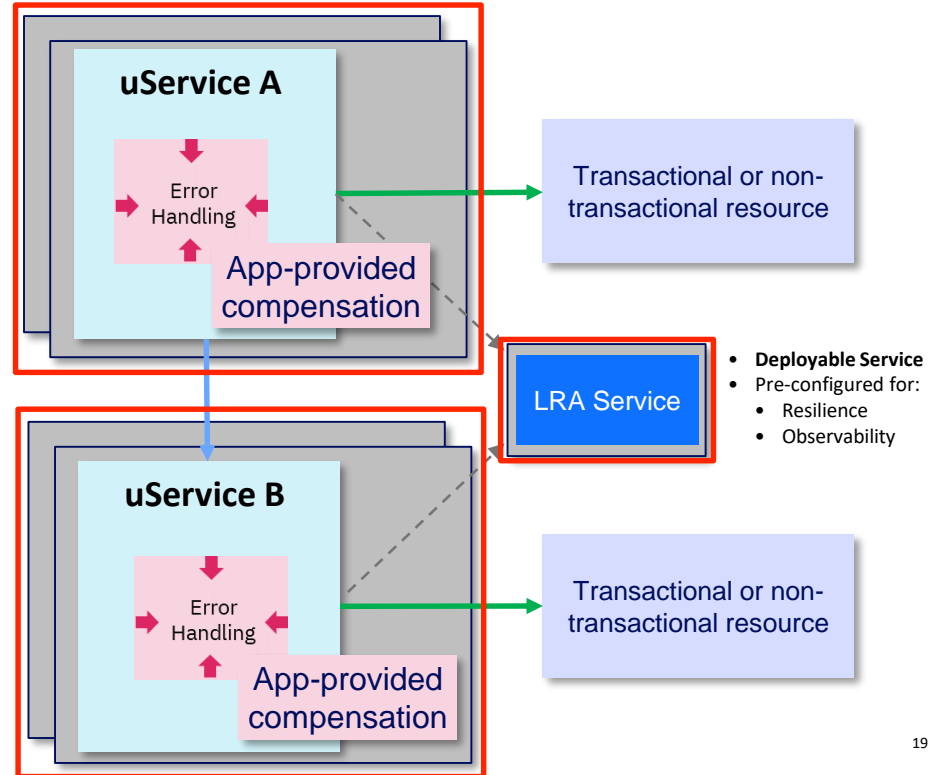
Accelerator for Cloud Native Transactions

Under Development – Extending IBM Cloud Pak for Apps V4.2 Accelerators
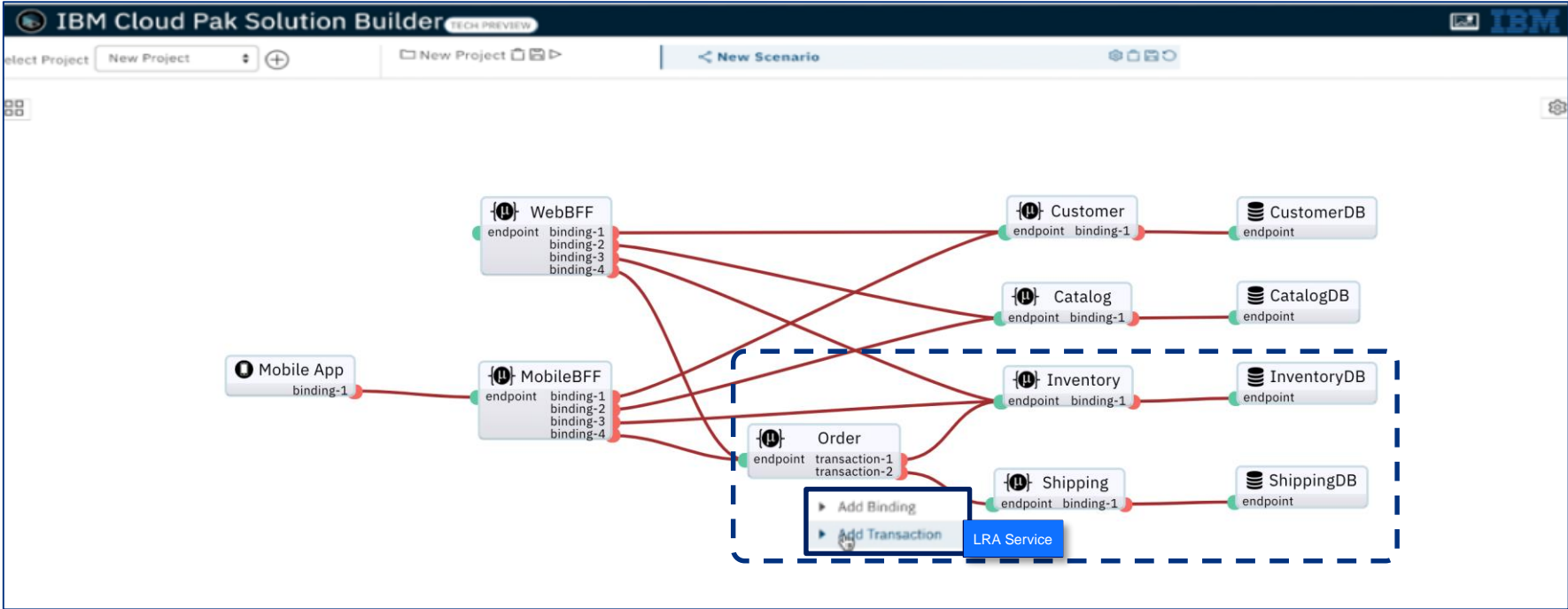
Polyglot SAGAs

Enabling the use of your language and framework of choice

Expand ecosystem to enable other languages/frameworks to participate in transactions

— Provide application SDK beyond Java MicroProfile annotations

— Integrate with non-Java application stacks for Transaction Accelerator

- **Application Stack**
- Pre-configured for:
  - Manager use
  - Resilience
  - Observability
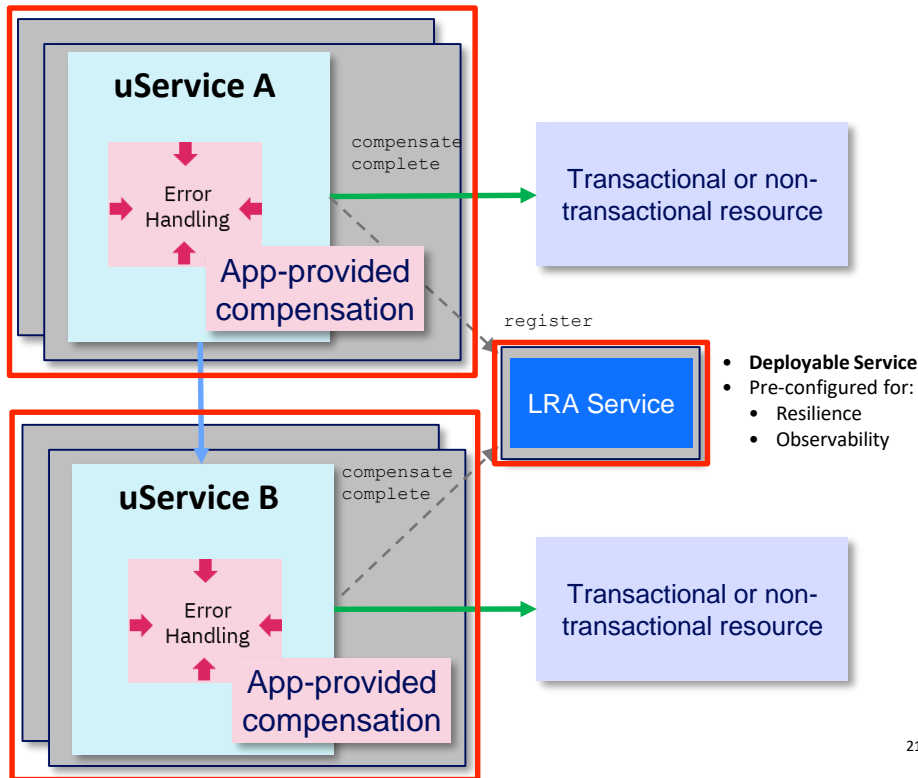
- **Application Stack**
- Pre-configured for:
  - Manager use
  - Resilience
  - Observability

uService A

Error Handling

App-provided compensation

compensate complete

Transactional or non-transactional resource

register

LRA Service

- **Deployable Service**
- Pre-configured for:
  - Resilience
  - Observability

uService B

Error Handling

App-provided compensation

compensate complete

Transactional or non-transactional resource

Join the CAB

WebSphere and Cloud Pak for Applications

Customer Advisory Board

claudiab@us.ibm.com
https://ibm.webex.com/meet/claudiab
http://ibm.biz/WASCABCommunityResources
http://ibm.biz/WebSphereAdvisoryBoard
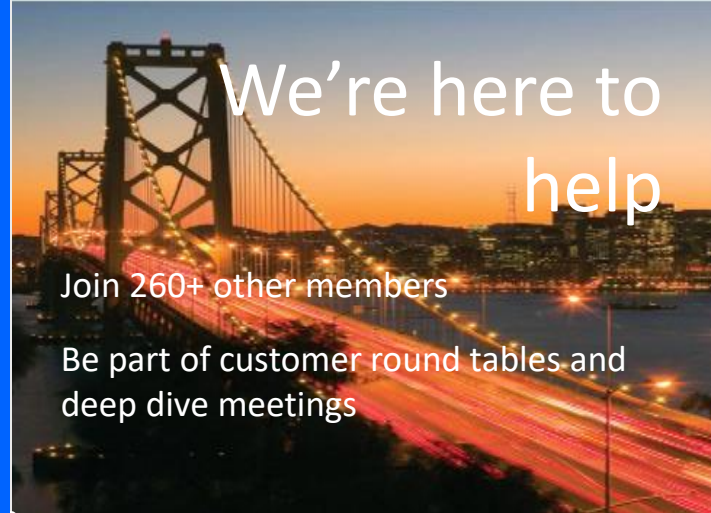
We're here to help

Join 260+ other members

Be part of customer round tables and deep dive meetings

Weekly meetings	**Monthly meetings**	**Other Programs**
Thursday and Friday	Business Partner track	Cloud Pak Week
9:15 am EST	Time zone friendly sessions	Previews
JOIN	JOIN	Demos
		Labs, workshops
		1-on-1

Engage when you have time:

- ✓ Stay in the loop at meetings
- ✓ Share solutions and pain points
- ✓ Connect with other customers
- ✓ Access to resources and experts
- ✓ Customized meetings
- ✓ Special offers

Notices and disclaimers

Notices and disclaimers continued

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.
Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Thank you

Ian Robinson, CTO IBM App Platform