IBM z Systems

TPFUG – MongoDB Interface to z/TPF Data

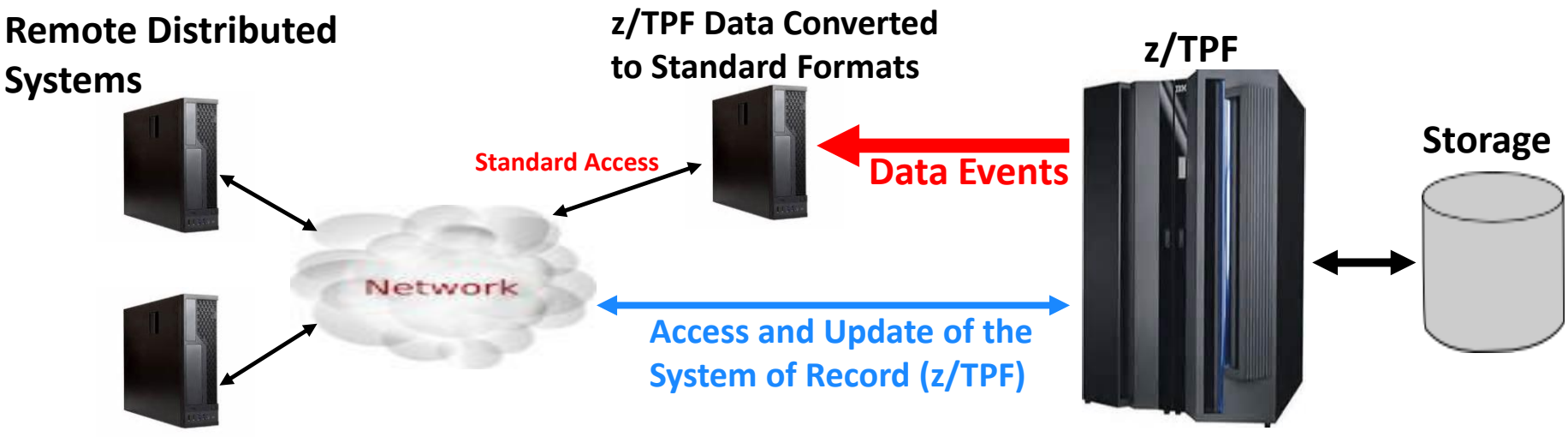
Jamie Farmer, TPF Development Lab

March 24, 2015

Disclaimer

- Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

How Does a Remote Platform Access z/TPF Data?



- Data events does allow you to push data to other platforms.
  - However, this does not satisfy the need for real-time access or update capabilities.

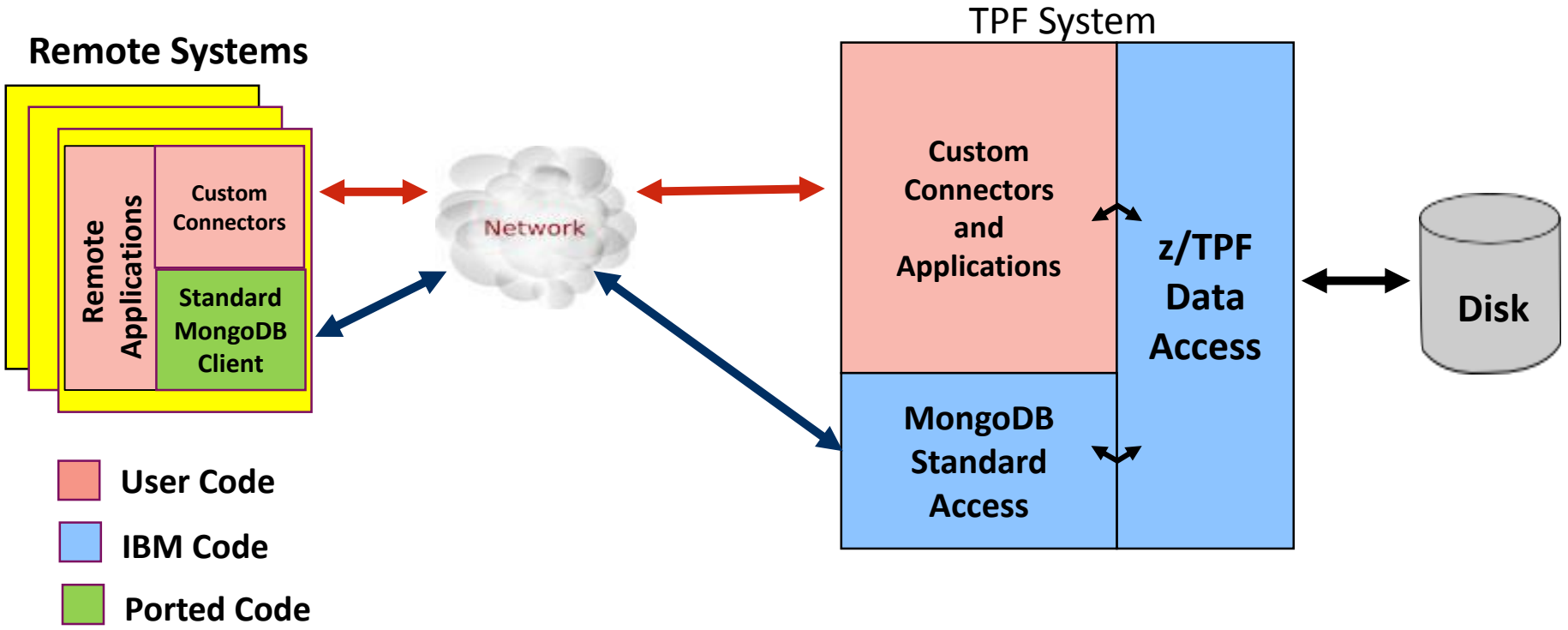
What is the Problem Being Solved?

- Today, remote access to data residing on z/TPF requires application code on z/TPF and on the remote platform.
  - Custom connectors into z/TPF
  - Parsing input request / Formatting responses
  - Database Access, for example z/TPFDF calls
- The time to market for new requirements needing access to the z/TPF data is usually dependent on z/TPF application changes to support it.
  - Cost of maintaining and enhancing these custom applications is significant.
    - What if the database changes?
    - What if the data requested by the end user changes?

What is MongoDB?

- Document oriented database supported by multiple platforms.
  - Data is organized as **documents** rather than organized in table based structures that you would find in a relational database.
  - Documents are contained within MongoDB **collections**.
  - Documents can be indexed for fast retrieval of a particular document.
- Classified as a NoSQL database.
  - Does not guarantee consistency of the data across multiple partitions.

Providing Standard Access to Data on z/TPF



- Provides standard, easy to use access to z/TPF data from remote systems using the MongoDB query language.
  - **No z/TPF application code required for access**
- Data residing on z/TPF can be accessed, consumed and updated by standard MongoDB clients on remote platforms

The MongoDB Interface to z/TPF

- **NOT** an implementation of a MongoDB server on TPF.
  - Provides an interface to existing z/TPF databases.
    - Cannot create new databases, add new fields, etc.
  - The z/TPF database still adheres to the ACID (Atomicity, Consistency, Isolation, and Durability) principles of OLTP transaction processing.
- Initial release will support z/TPFDF databases
  - Current plan is to support other z/TPF databases (find/file) in future deliverables

Why We Chose MongoDB?

- z/TPF data described in document form seems to be a perfect fit.
- Similar indexing concepts between z/TPF and MongoDB.
- Engrained in the industry and widely used.
  - MongoDB clients exist in almost any language.
  - Arguably the most popular document store in the industry.
- In addition to accessing data programmatically.
  - Open tooling exists like GUIs and shell based front ends to access and update data.
    - For example Robomongo (robomongo.org)

The Magic of the MongoDB Interface

- The magic of MongoDB is in the database description.
  - User data format of logical records described in DFDL.
  - XML file describing the z/TPFDF Database Definition (DBDEF).
- z/TPF command driven tooling is provided to create the initial database description.
  - Optionally customize it to provide meaningful names to z/TPFDF files, paths, records, and fields.
- Database descriptions can be created on a z/TPFDF file basis.
  - Loading the database description for a particular z/TPFDF file to the z/TPF system, enables that z/TPFDF file to be accessed through the MongoDB interface.
- For more information on describing your database and DFDL.
  - IBM DFDL Presentation in the Database subcommittee.
  - IBM education session on DFDL is being held Wednesday morning.

MongoDB Constructs Mapped to z/TPFDF

- In normal MongoDB after connecting to the server the remote client accesses a **database**, provides a **collection** name and retrieves and updates **documents** within that **collection.**
- The following table maps MongoDB terminology to z/TPFDF.

MongoDB Terminology	z/TPF Terminology
Database	"tpfdf" – Only database supported in initial deliverable
Collection	z/TPFDF File
Document	z/TPFDF subfile
Index	Defined z/TPFDF Path statement

z/TPF MongoDB Interface Functionality

- Create new documents
  - Includes indexing the new document.
- Retrieve existing documents
  - Filter results based on user supplied criteria (ie. All history records).
- Update existing documents
  - Replace an entire document
  - Update the indexes of a document
  - Insert and remove records within a document
  - Update specific fields within records of a document
- Delete existing documents
  - Includes deindexing the document being deleted

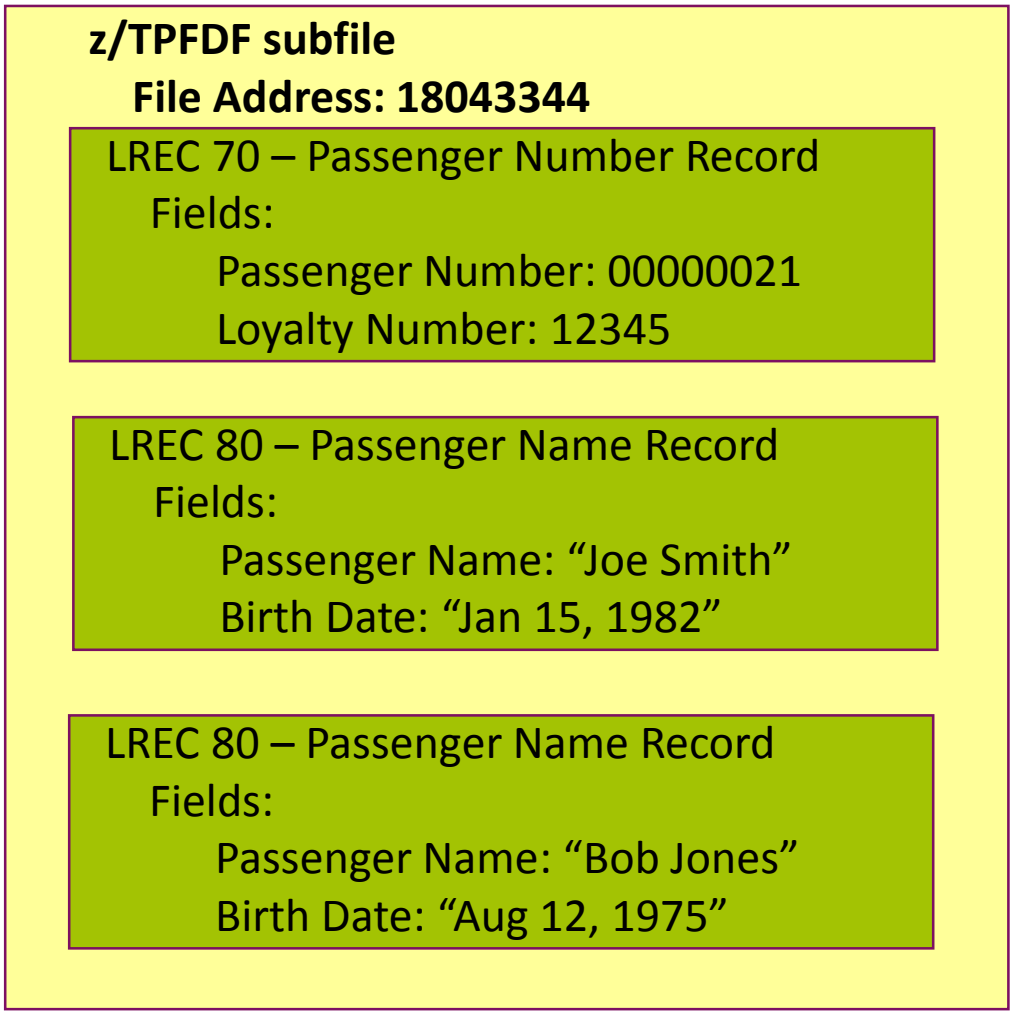
z/TPF MongoDB Interface Functionality (Continued)

- A MongoDB server provides user security using MongoDB standard APIs
  - z/TPF will allow you to authenticate users through the same MongoDB APIs
  - Controlling which users are allowed to access SSUs, Collections, etc.
- Provide an SSL version of the z/TPF MongoDB Interface
- Ability to analyze performance of the z/TPF MongoDB Interface

z/TPF MongoDB Interface Integrity

- z/TPF MongoDB Interface Database Integrity
  - Just like the standard MongoDB server, update operations against the z/TPF MongoDB Interface are atomic
    - Either the entire update is applied, or the database is unchanged.
- z/TPF MongoDB Interface System Integrity
  - Cannot hold database locks across MongoDB requests from a remote system
  - Can only search for documents through defined indexes to prevent a single request from traversing the entire database
    - Potentially generating millions of I/Os

An Example of a z/TPFDF Subfile



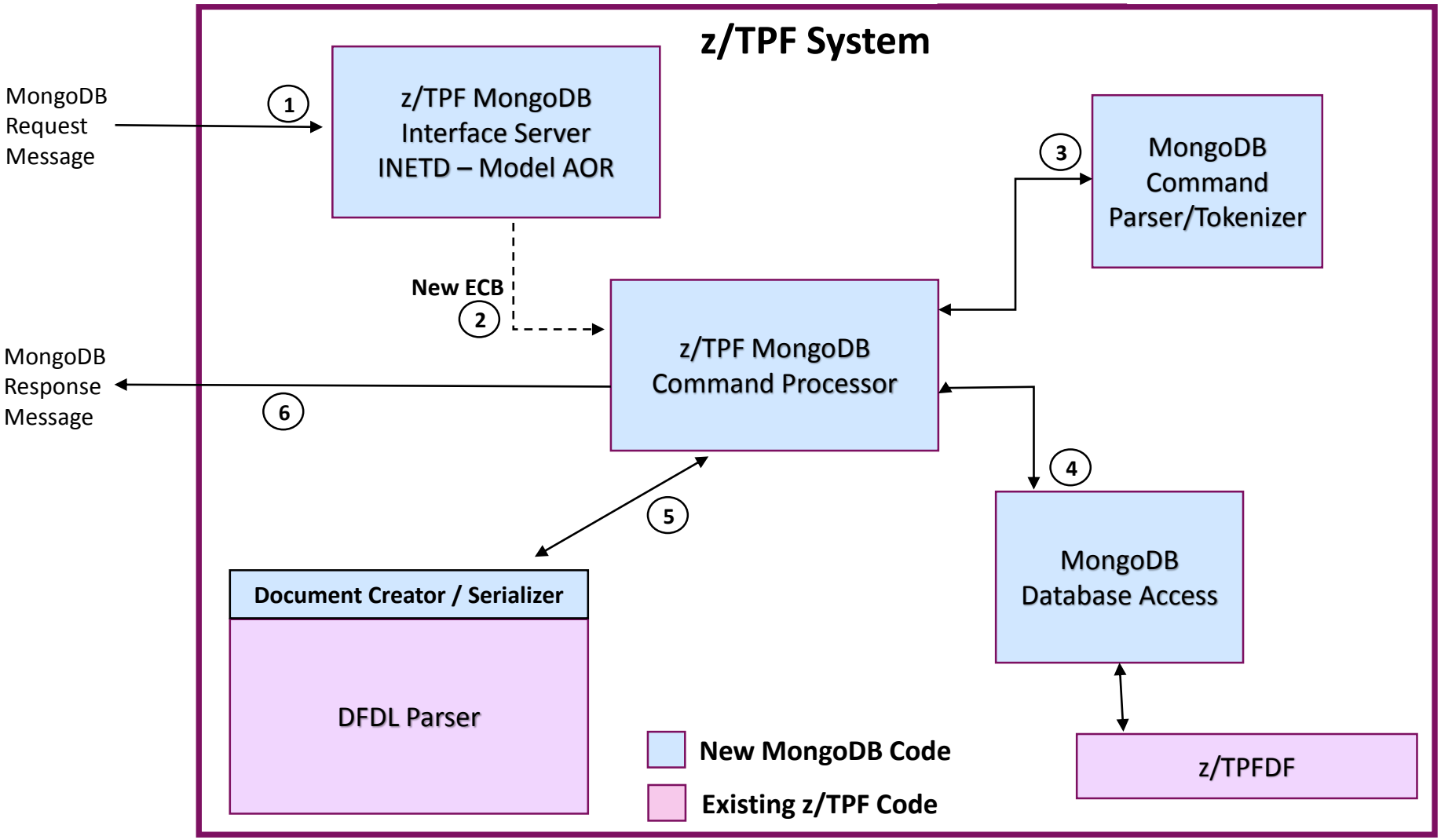
- z/TPFDF subfile contains three records
  - One Passenger Number Record
  - Two Passenger Name Records
- File address of the prime block of the subfile is 18043344

Example of a Document Representing the z/TPFDF Subfile

```
{
 "PassengerNumberRecord" : [
        {
            "PassengerNumber" : 00000021
            "Loyalty Number" : 12345
        }
    ],
 "PassengerNameRecord" : [
        {
            "PassengerName" : "JOE SMITH "
            "Birthdate" : "Jan 15, 1982"
        }
        {
            "PassengerName" : "BOB JONES "
            "Birthdate" : "Aug 12, 1975"
        }
    ],
"_id" : ObjectId("000000000000000018043344")
}
```

- Documents are built based on the records contained within the z/TPFDF subfile.
- Each record is represented as an array
- Each document contains an _id field which allows for direct access of the document in the future.
  - File address of prime block
  - Best performance to access document in the future

How Does it Work?
Retrieving a Document Message Flow



Retrieving a Document - Details

1. MongoDB request received to z/TPF from the remote distributed system.
2. The Internet Daemon (INETD) creates a new ECB to service the request.
3. The MongoDB command processor parses and tokenizes the request, determining the type of command being received.
4. The MongoDB database access layer is invoked to retrieve the target document (sub-file).
5. The DFDL parser is used to build a document from the binary (z/TPFDF) data.
6. The response document is sent back to the remote distributed system.

A Call for Sponsored Users

- Looking for sponsored users of the z/TPF MongoDB Interface.
  - Participate in delivery playbacks.
  - Provide feedback on implementation and priority of the implementation.
  - Optionally participate in Beta of the z/TPF MongoDB Interface.
- Please contact IBM for more information.

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., 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.
- *(Include any special attribution statements as required – see Trademark guidelines on https://w3-03.ibm.com/chq/legal/lis.nsf/lawdoc/5A84050DEC58FE31852576850074BB32?OpenDocument#Developing%20the%20Special%20Non-IBM%20Tr)*

Notes

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.