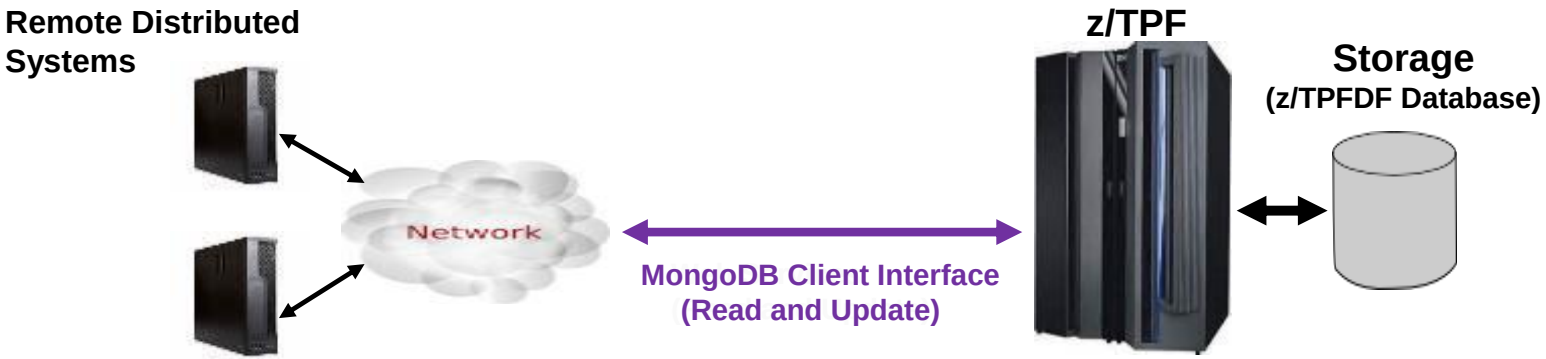
z/TPF Support For MongoDB Enhancements

Jamie Farmer
Database Subcommittee

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.

MongoDB Access to z/TPF Data



- Can access and update z/TPF data using a standard, unmodified MongoDB client on another platform.
  - Just point the MongoDB client to the z/TPF system!
  - No z/TPF application updates required
- z/TPF MongoDB User Security allows you to administratively control access to z/TPFDF data based on user ID accessing it.

Agenda

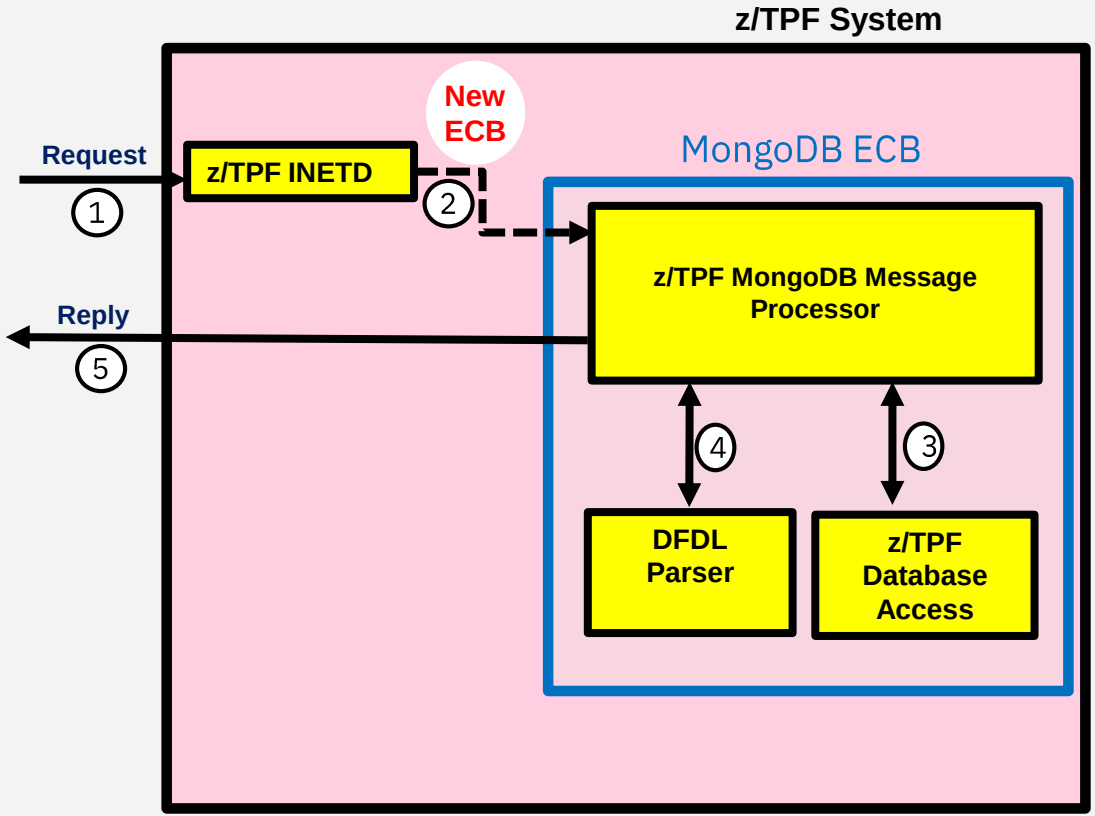
- A user can apply “user specific” processing when z/TPF processes a MongoDB request.
- A user can configure z/TPF to detect idle or abandoned MongoDB sessions.
- A user can filter data within a MongoDB collection (z/TPFDF file), providing added security and control of data presented to remote users.

A user can apply “user specific” processing when z/TPF processes a MongoDB request.

PJ44525

z/TPF MongoDB Message Flow

- z/TPF MongoDB requests are processed without any z/TPF application code required.
- User specific processing may be needed on each MongoDB request

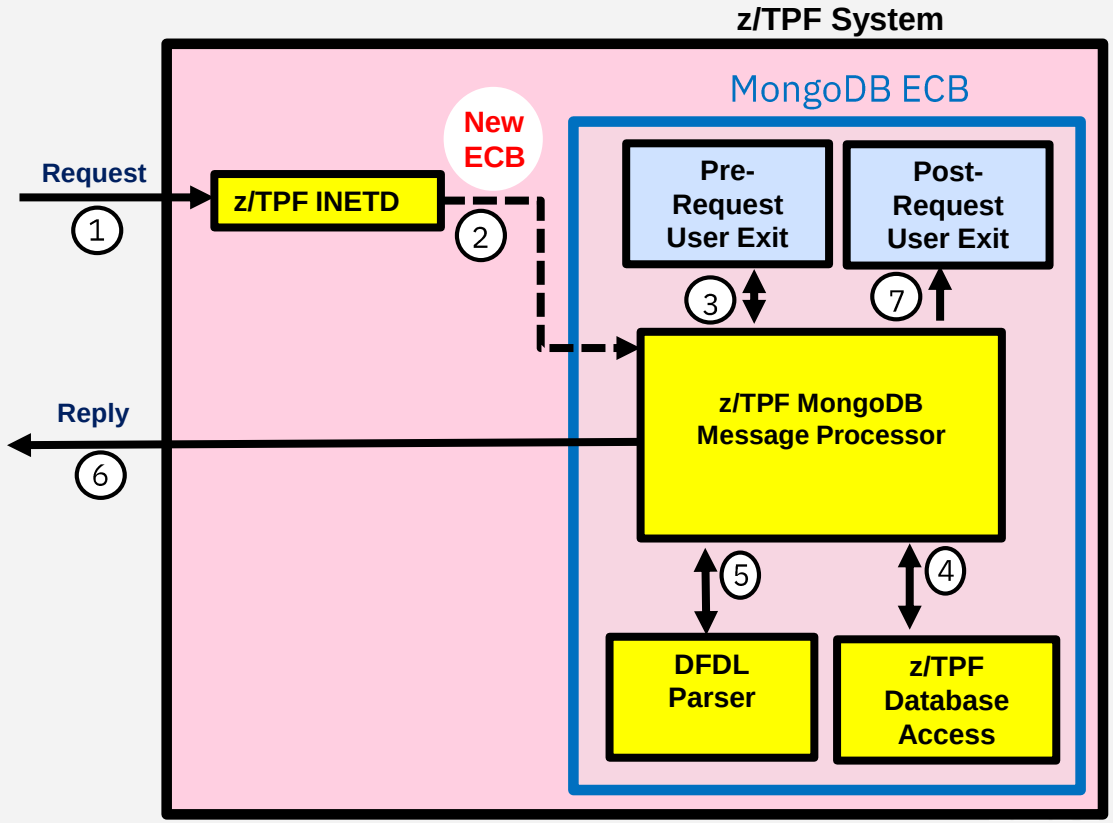


User Specific Processing on Each MongoDB Request Examples

- A user can make additional resource checks before processing z/TPF MongoDB requests.
  - Resource checking may be done based on user issuing request
- A user can make additional state checks to prevent MongoDB requests below NORM state
- A user can track user's requests and resources consumed for accounting purposes.

z/TPF MongoDB Message Flow With User Exits

- Two new user exits have been created:
  - Pre-request user exit is called before processing a given MongoDB request.
  - Post-request user exit is called after processing is complete before sending the response.



z/TPF MongoDB Pre-Request User Exit

- Inputs
  - Socket Descriptor of Session
  - Indicator if session is using SSL
  - Username associated with request
- Output
  - What system should do with request?
    - Allow request (Default Return Code)
    - Reject Request (including error text for remote client)
    - Reject and Tear Down Session (including error text for remote client)

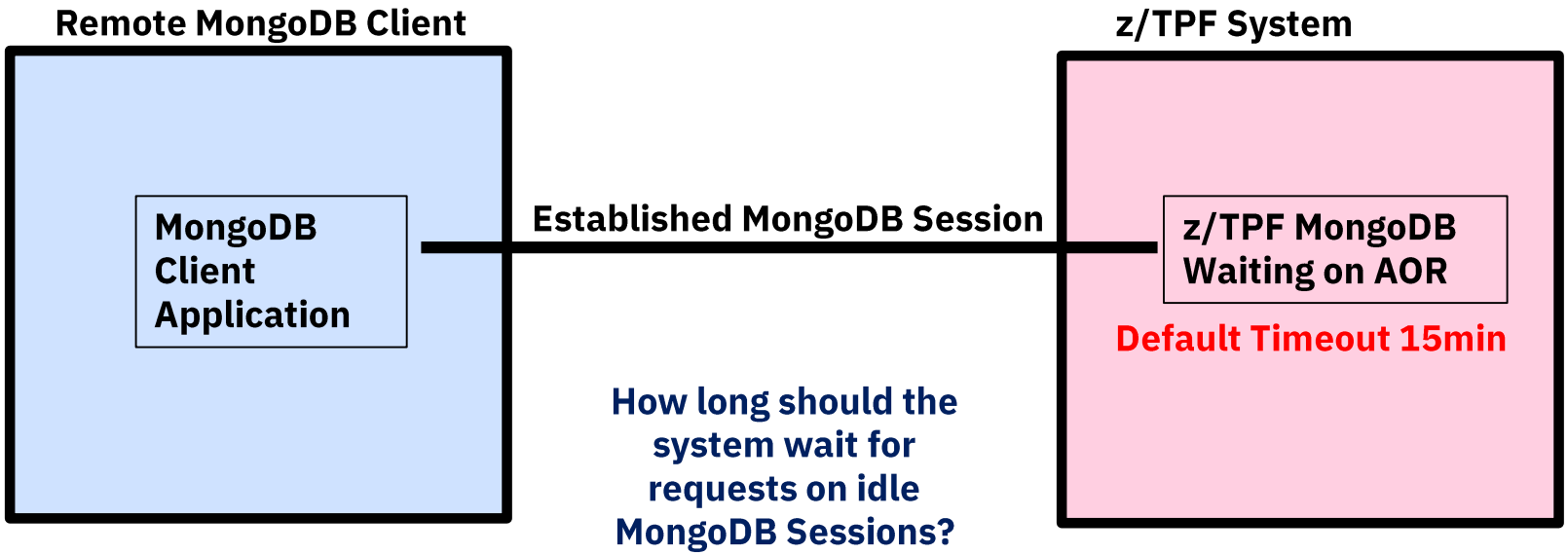
z/TPF MongoDB Post-Request User Exit

- Inputs
  - Socket Descriptor of Session
  - Indicator if session is using SSL
  - Username associated with request
  - Request type (find, update, insert, etc.)
  - Database
  - Collection
  - Indicator if a response is sent
  - Indicator if request was successful
- Output
  - None

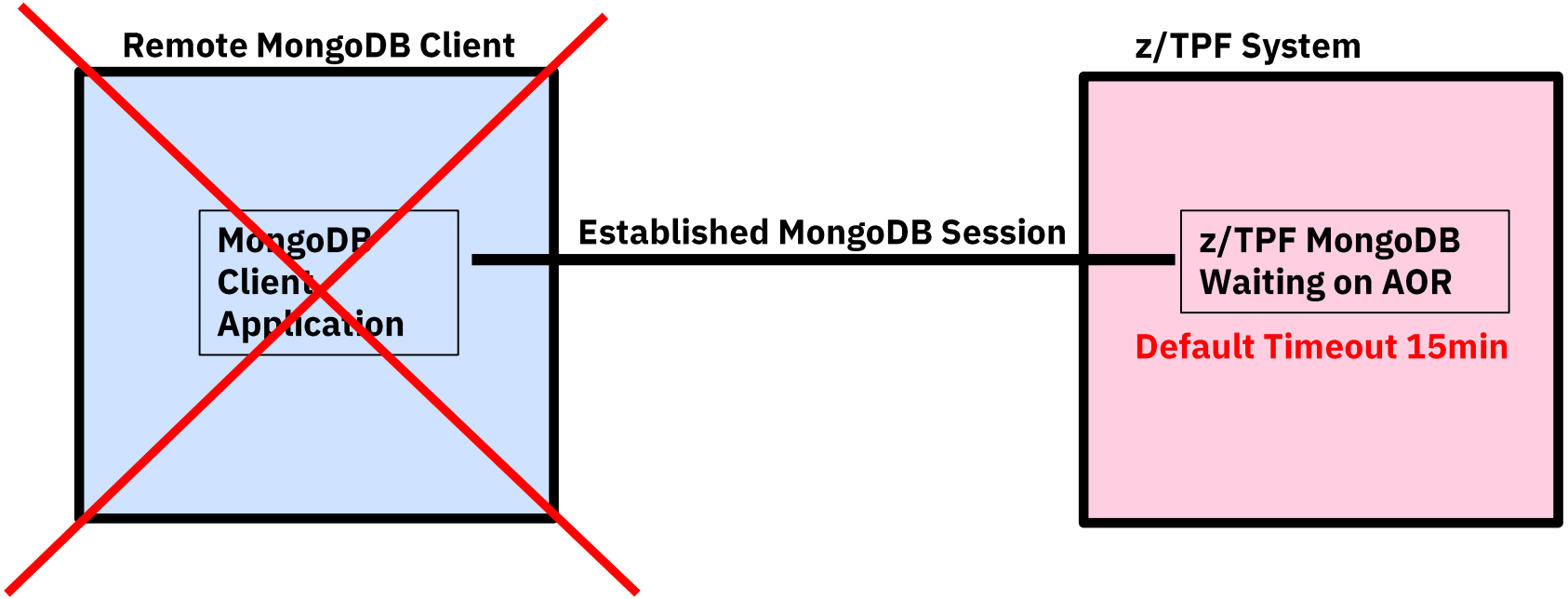
A user can configure z/TPF to detect idle or abandoned MongoDB sessions.

PJ44525

z/TPF MongoDB Idle Sessions



z/TPF MongoDB Remote Failure



How does the z/TPF system detect remote failures?

z/TPF MongoDB Detecting Idle Sessions

- A user can configure how long (in minutes) to keep idle z/TPF MongoDB session open
  - The default timeout is 15 minutes if not specified
  - Defined using timeout option on the INETD definition for MongoDB

zinet add s-mongo model-daemon pgm-cads xparm- --auth --timeout 3

- A timeout value of zero means the system will not timeout idle sessions.

z/TPF MongoDB Detecting Remote Failures

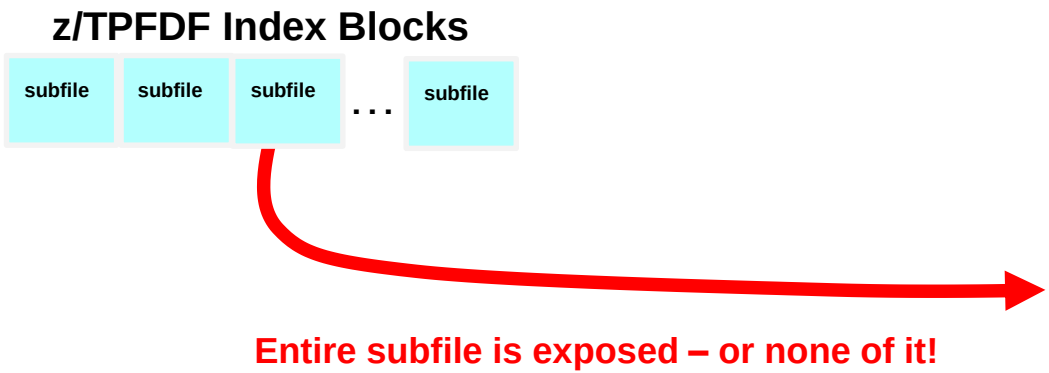
- TCP keepalive has been enabled for z/TPF MongoDB sessions to detect remote failures
  - Frequency of keepalive messages is controlled at the system level using:
    - ZNKEY TCPALIVE-xx

A user can filter data within a MongoDB collection (z/TPFDF file), providing added security and control of data presented to remote users.

PJ44688

Representation of Subfile as Document Example

The original MongoDB support correlates a z/TPFDF subfile to a MongoDB document




```
{ “CustomerNameRecord" : [
      { “custID" : "00000021“,
        “name” : “Joe”}
           { “custID" : "00000022“,
       “name” : “Bob” }
]
{ “CustSSNRecord" : [
           { “custID" : "00000021“,
       “SSN" :  12345 }
           { “custID" : "00000022“,
            “SSN" :  54321}
],
{ “CustAddress” : [
           { “custID" : "00000021“,
       “address” : “1 Oak Ln.”}
           { “custID" : "00000022“,
            “address” : “1 Elm Ln.”}
]
```

Filtered Collections

- Filtered collections provide the ability for a database administrator to define MongoDB collections that only exposes a subset of the data contained in them.
  - Implemented for security reasons, but can be used to control what data is presented to remote user for performance (send smaller payloads!).

- Defined like other MongoDB collections through the z/TPFDF collection descriptor (adbi.xml)

Defining a Collection As A Filtered Collection

Traditional MongoDB Collection Definition

```
<tns:Collection name="AirCoCust"
    dfdlfile="AirCoCust.tpfdf.dfdl.xsd" collectionId="C412"
    ……
    <tns:Lrecs>
        <tns:Lrec name="CustomerRecord" id="70" />
        <tns:Lrec name="PaymentRecord" id="80" />
        <tns:Lrec name="ServiceRecord" id="C0" />
    </tns:Lrecs>
</tns:Collection>
```

Traditional collections require all LRECs to be defined

Filtered MongoDB Collection Definition

```
<tns:Collection name="AirCoNoPayment" filtered="true"
    dfdlfile="AirCoCust.tpfdf.dfdl.xsd"
collectionId="C412"
    ……
    <tns:Lrecs>
        <tns:Lrec name="CustomerRecord" id="70" />
        <tns:Lrecname="ServiceRecord" id="C0" />
    </tns:Lrecs>
 </tns:Collection>
```

Filtered collections allow you to pick which records you want included

Additional Filtered Collection Use Cases

- There is a handful of customer data contained in a given z/TPFDF file
  - For example, a z/TPFDF file may contain a dozen different airline carrier information.
  - A separate filtered collection can be created for each airline carrier with filters using fixed data
  - Existing MongoDB user security allows you to control which users can see a given carrier's data.

- There are thousands or even millions of different customer's data in a single z/TPFDF subfile.
  - For example, a z/TPFDF file may contain a million different customer's credit card information
  - A single filtered collection can be created with filters using variable data (the data supplied by users to locate a subfile (or document)
  - A user exit will be put in place to allow you to control which users can see which credit card information (ie. various banks)

Filtering Using Fixed Data Within Records

Traditional MongoDB Collection Definition

```
<tns:Collection name="AirCoCust"
    dfdlfile="AirCoCust.tpfdf.dfdl.xsd" collectionId="C412"
    ……
    <tns:Lrecs>
        <tns:Lrec name="CustomerRecord" id="70" />
        <tns:Lrec name="PaymentRecord" id="80" />
        <tns:Lrec name="ServiceRecord" id="C0" />
    </tns:Lrecs>
</tns:Collection>
```

Filtered MongoDB Collection Definition

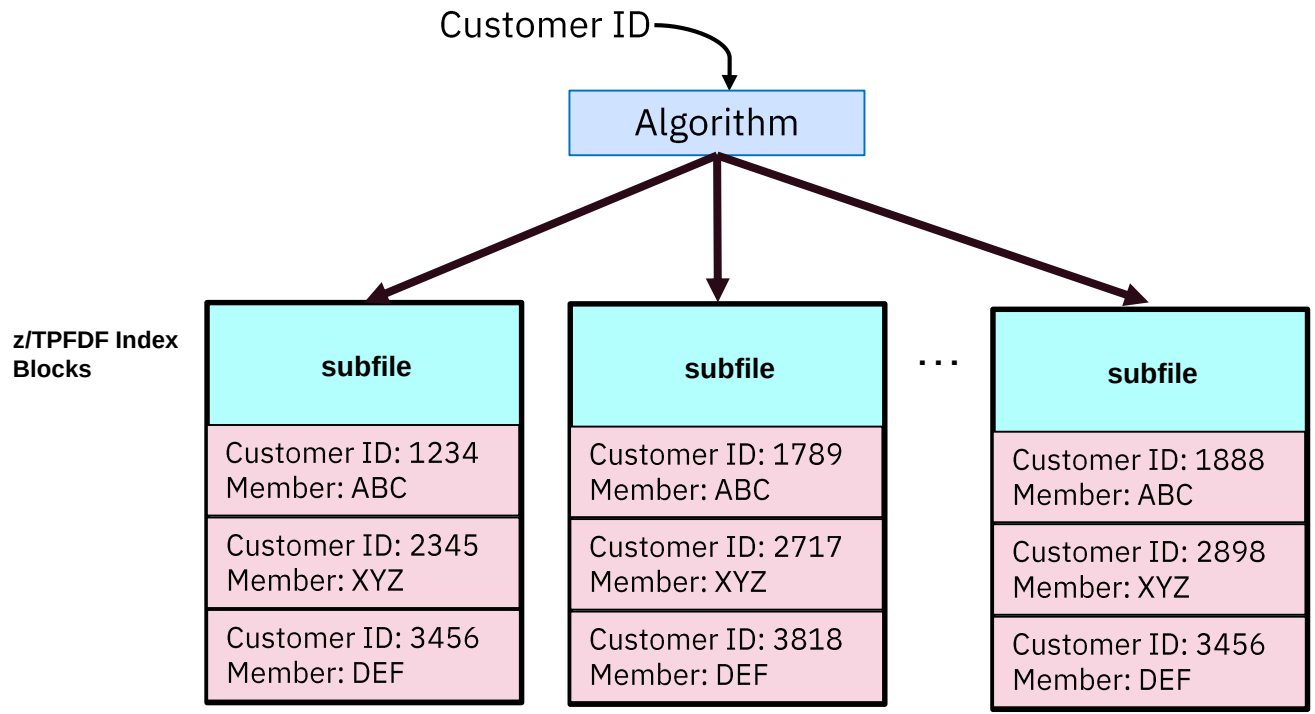
```
<tns:Collection name=" AirCoCustZZ" filtered="true"
    dfdlfile="CustInfo.tpfdf.dfdl.xsd" collectionId="C412"
    ……
    <tns:Lrecs>
        <tns:Lrec name="CustomerRecord" id="70" />
            <tns:FilterRule>
                {./carrierID eq 'ZZ'}
            </tns:FilterRule>
        <tns:Lrec name="ServiceRecord" id="C0" />
            <tns:FilterRule>
                {./carrierID eq 'ZZ'}
            </tns:FilterRule>
    </tns:Lrecs>
</tns:Collection>
```

Traditional collections do not allow filtering of records.

Filtered collections allow you to filter records based on the contents of the record.

- Uses embedded DFDL expression for fields in a record allowing for complex expressions

Example Of Millions of Customers In z/TPF File



- Millions of customers are distributed across a set of fixed files
- Cannot create a separate filter for every customer record.
- Any filter would have to based on the user input - customerID

Filtering Using Variable Data From Remote Query

```
<tns:Collection name="AirCoCust" filtered="true"
    dfdlfile="AirCoCust.tpfdf.dfdl.xsd" collectionId="C412"
    ……
    <tns:Indexes>
        <tns:Index name="CustomerByID" number="0" >
            <tns:VariableDefs>
                <tns:Variable>
                    <tns:name>custIDVar</tns:name>
                    <tns:field>custID</tns:field>
                </tns:Variable>
            </tns:VariableDefs>
        </tns:Index>
    </tns:Indexes>

    <tns:Lrecs>
        <tns:Lrec name="CustomerRecord" id="70" >
            <tns:FilterRule>
                {./custID eq $custIDVar}
            </tns:FilterRule>
        </tns:Lrec>
        ……
    </tns:Lrecs>
</tns:Collection>
```

- The index definitions map the z/TPFDF paths used to find a given subfile
  - Define variables based on the input
  - Variables are setup when subfile is found through that index.

- The logical record filters can be defined to present a record if the field of a record matches the variable
  - Uses '$' in DFDL expression to indicate varNum is a variable.

Filtered Collections and Security

- Access to filtered collections is controlled through the z/TPF MongoDB user security
  - User: AirUser has read access to the **CustNoPayment** collection
  - Any other collection is restricted unless authorization is given.

- When filtering collections based on user supplied input, multiple users can access different data based on the input.
  - Existing user security package does not prevent malicious users from accessing other users data.

MongoDB Filtered Collection Access Control User Exit

- The UATH_filteredCollection user exit in uath.cpp is called when a user attempts to access a filtered collection.
  - Allows you to prevent users from accessing data within a collection that they are not allowed to see.
  - For example, if a user owning credit card number 1234 tries to access credit card 4321.
- Use this user exit when the existing MongoDB user security package cannot be used.
  - For example when multiple filtered collections can not be created

Summary

A user can apply “user specific” processing when z/TPF processes a MongoDB request.	**PJ44525 (PUT 14)**
A user can adjust how long a MongoDB session remains open waiting to process a request.	**PJ44525 (PUT 14)**
A user can filter data within a MongoDB collection (z/TPFDF file), providing added security and control of data presented to remote users.	**PJ44688 (PUT 14)**

Thank You!

Questions?

Trademarks

IBM, the IBM logo, ibm.com and Rational 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.

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.