SERVICE BROKER: The Queue Continuum

ED LEIGHTON-DICK

A LITTLE ABOUT ME…

- 22 years in IT
- SQL Server since 2000
- DBA, 2006-present
- Chapter Leader, I-380 PASS
- MCITP

WHAT WE'LL COVER

- What is Service Broker?
- How do I configure and use it?
- Troubleshooting Service Broker
- A tale of two systems: GreatAmerica Financial Services

WHAT IS SERVICE BROKER?

WHAT IS SERVICE BROKER?

- Queue-based messaging structure
  - Way to pass messages between processes

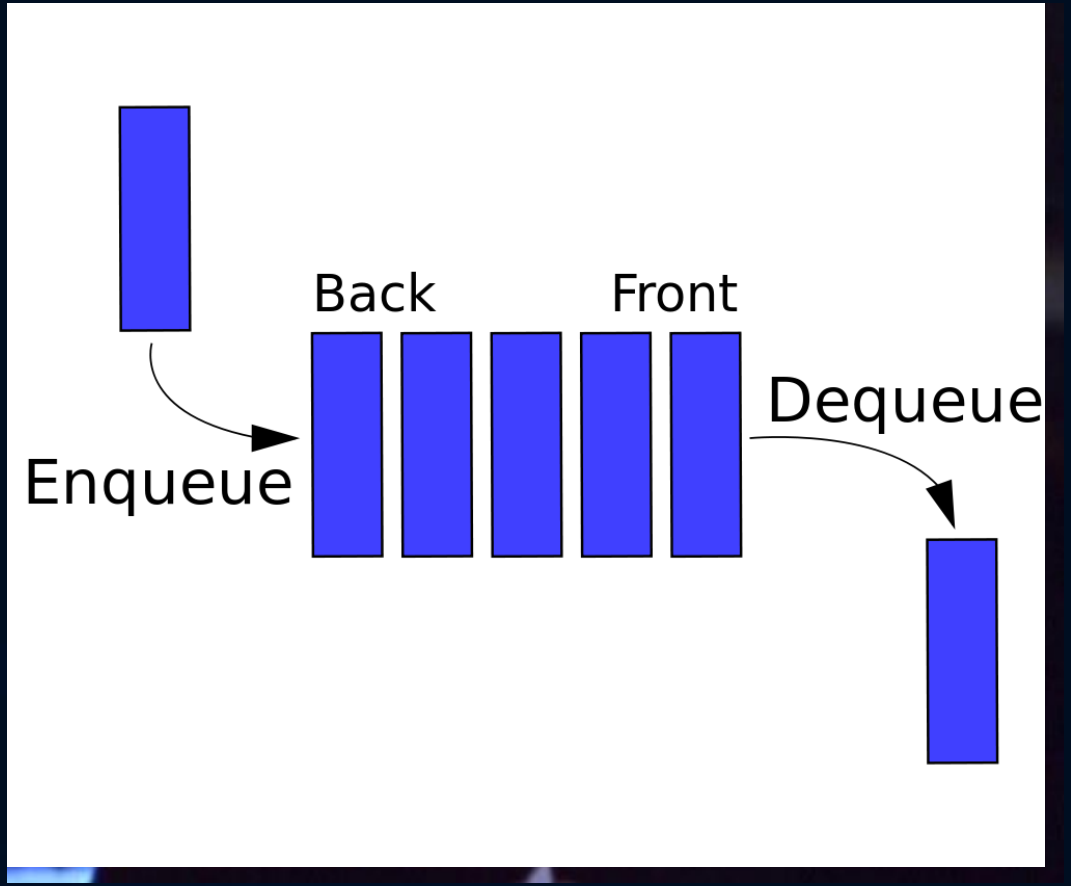


Image credit: wikipedia.org

WHAT IS SERVICE BROKER?

- Queue-based messaging structure
- Asynchronous

WHAT IS SERVICE BROKER?

- Queue-based messaging structure
- Asynchronous
- Distributed
  - Can work within a single instance or across multiple instances

WHAT IS SERVICE BROKER?

- Queue-based messaging structure
- Asynchronous
- Distributed
- Scalable
  - “Multithreaded” queue processing

WHAT IS SERVICE BROKER?

- Queue-based messaging structure
- Asynchronous
- Distributed
- Scalable
- Resilient
  - Guaranteed delivery
  - Transactional
  - Persistent

USES FOR SERVICE BROKER

- SQL Server Event Notifications
- SQL Server Database Mail

USES FOR SERVICE BROKER

- SQL Server Event Notifications
- SQL Server Database Mail
-

Bad	Good
• Point-of-sale systems	• Emailing a receipt • Order fulfillment
• File submissions	• File processing
• Error handling	• Error notifications

USES FOR SERVICE BROKER

- SQL Server Event Notifications
- SQL Server Database Mail
- Applications where immediate action is not required
- Multi-step processes
- Complex processes
- Distributed processes
- Resilient processes

HOW DO I CONFIGURE SERVICE BROKER?

COMPONENTS

- Required objects
- Sending and receiving messages
- Optional objects
- Error handling
- Troubleshooting

REQUIRED OBJECTS: MESSAGE TYPES

- Defines the type of data to be included in the message:
  - NONE – No validation
  - EMPTY – Must be NULL
  - WELL_FORMED_XML
  - VALID_XML WITH SCHEMA COLLECTION *xsdName*

REQUIRED OBJECTS: CONTRACTS

- Defines the message types to be used in the conversation
- Specifies which party can send each message type
  - INITIATOR
  - TARGET
  - ANY

REQUIRED OBJECTS: QUEUES

- Defines the storage for the queue
- One queue is required for each side of the conversation (INITIATOR, TARGET)

REQUIRED OBJECTS: SERVICES

- Defines the Service Broker service
- Relates the queue to the contracts it can process
- One is required for each side of the conversation

SENDING MESSAGES

- BEGIN DIALOG CONVERSATION – Opens the conversation
- SEND ON CONVERSATION – Sends a message

RECEIVING MESSAGES

- RECEIVE – Reads a message from the queue
- END CONVERSATION – Closes the conversation
  - Must be done on both the target and the initiator

OPTIONAL OBJECTS: ACTIVATION PROCEDURES

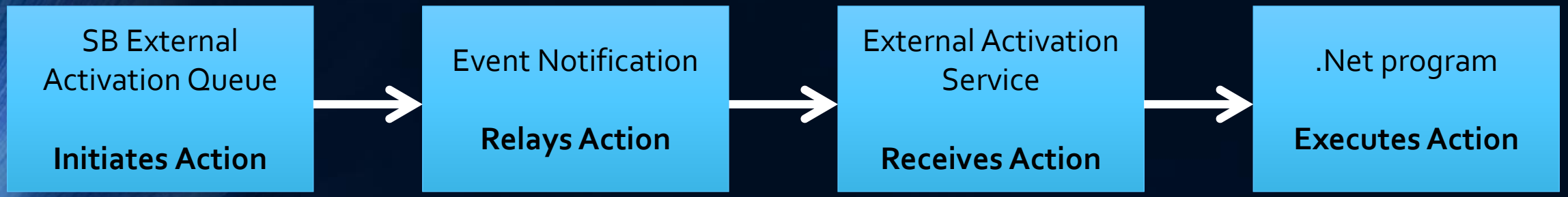
- Enables a Service Broker queue to handle messages automatically as they arrive
- Can enable multiple instances of activation procedure (“readers”) to simulate multithreading

OPTIONAL OBJECTS: ENDPOINTS AND ROUTES

- Enable Service Broker's distributed capabilities
- Required on each side of the connection
- Can be secured through either transport security or dialog security

OPTIONAL OBJECTS: EXTERNAL ACTIVATION

- Allows Service Broker messages to be processed outside of SQL Server
- Useful when the service program needs to execute processes that shouldn't or can't be executed within SQL Server (i.e., call a web service)



DEMO

HANDLING ERRORS

Example:
XML in first message contains an invalid character

BEGIN TRANSACTION

RETRIEVE TOP (1)

XML processing fails

ROLLBACK TRANSACTION

WRONG!

- **After 5 attempts,**
  SQL Server stops the queue!

HANDLING ERRORS

- Poison messages
  - Messages that exceed the maximum number of delivery attempts
- Ways to handle poison messages
  - Rollback the transaction and try again
  - Redirect the failed message
  - End the conversation with an error
  - Event notification

HANDLING ERRORS

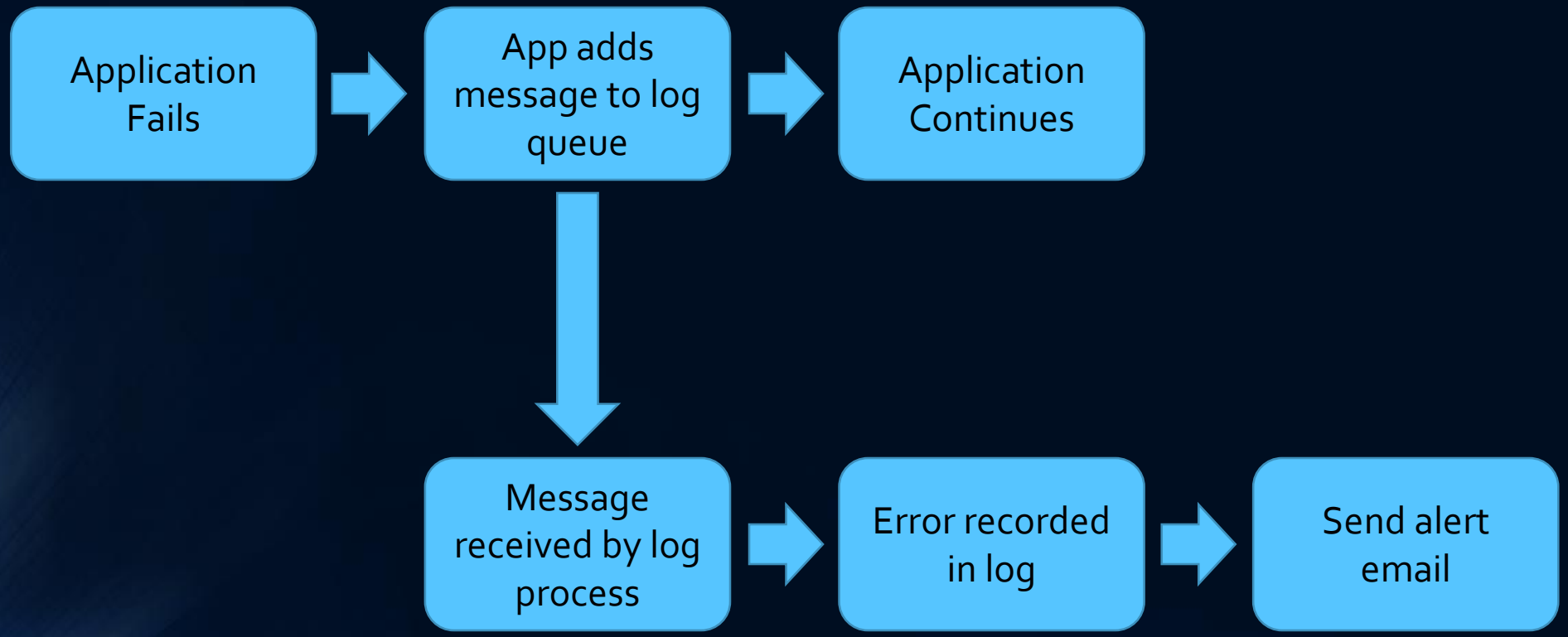
```
BEGIN TRY
    BEGIN TRANSACTION
    RETRIEVE TOP 1
    SAVE TRANSACTION haveMessage
    XML processing fails
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION haveMessage
    Redirect row and/or handle error
    COMMIT TRANSACTION
END CATCH
```

TROUBLESHOOTING QUEUES

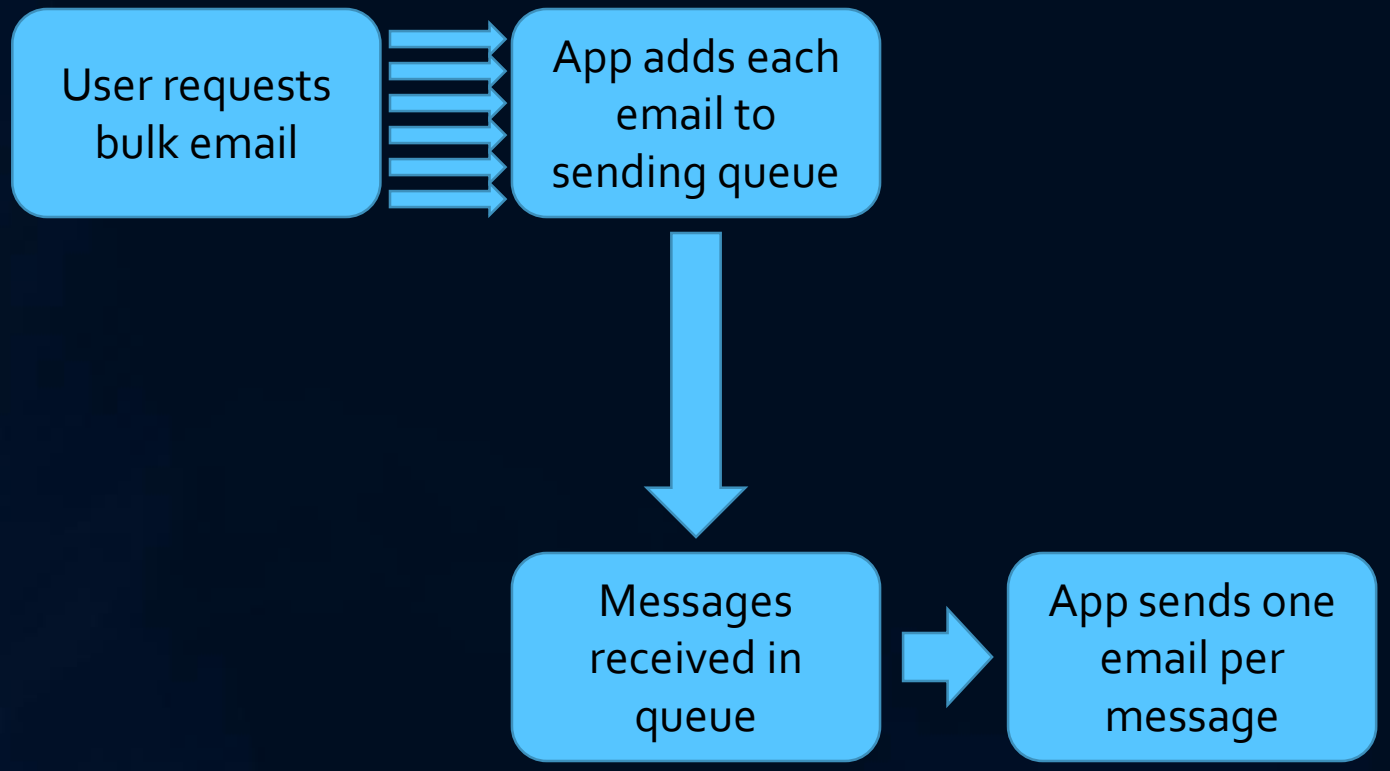
- Catalog views for Service Broker
  - sys.transmission_queue
  - sys.conversation_endpoints
- Dynamic Management Views (DMVs) for Service Broker
- SQL Server Profiler
- Extended Events (XEvents)/Event Notifications
- ssbdiagnose
- PerfMon

IDEAS AND CASE STUDIES

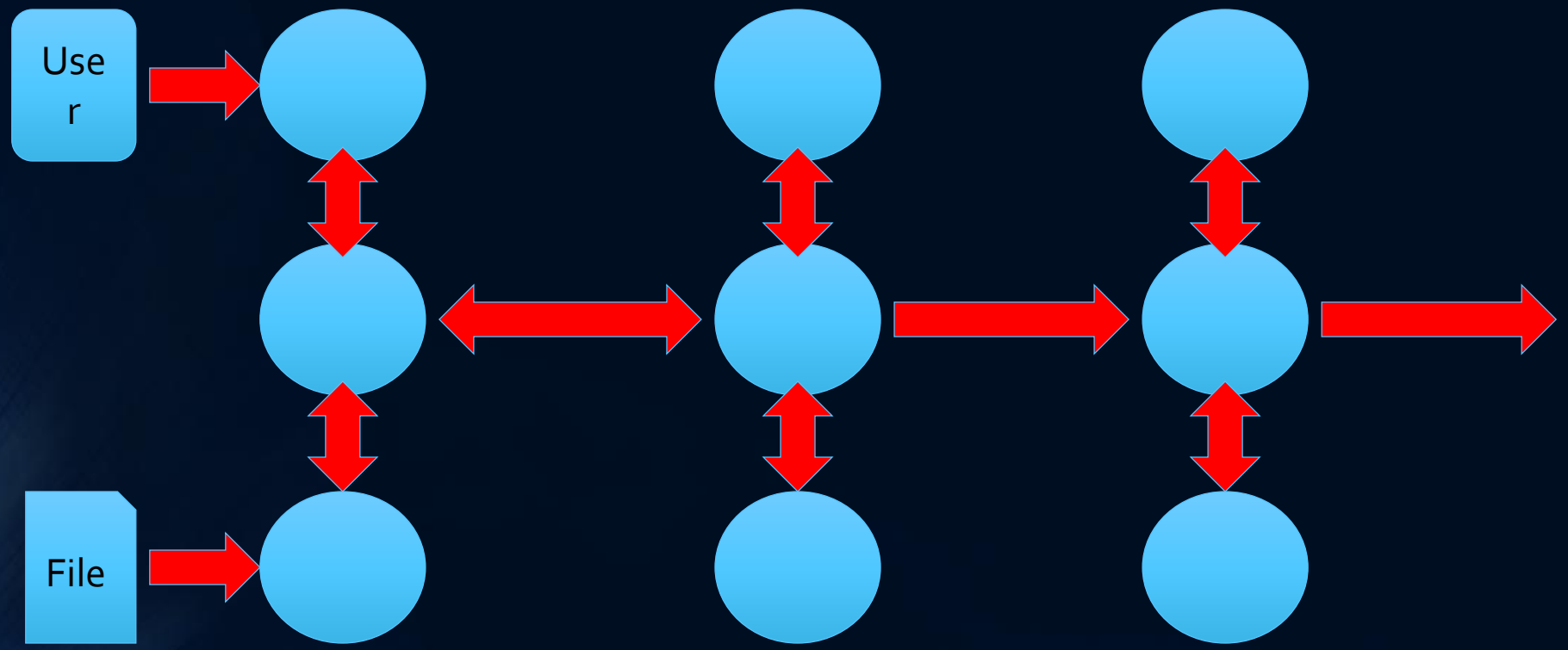
IDEA: ERROR LOGGING SYSTEM



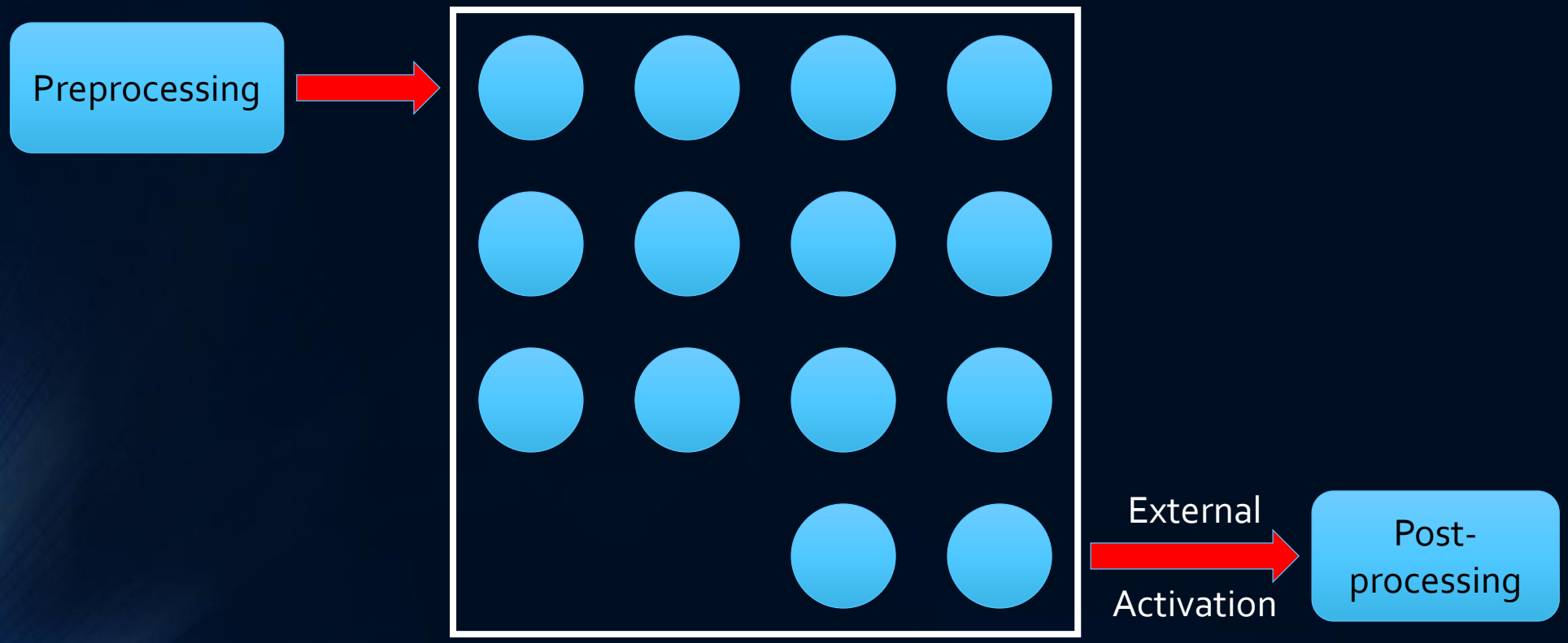
IDEA: BULK EMAILER



GREATAMERICA:
USAGE TRANSACTION SYSTEM (UTS)



GREATAMERICA: INVOICING SYSTEM



GREATAMERICA: LESSONS LEARNED

- Keep messages small
- Keep messages focused
- Queues should have a single purpose
- Messages should flow in a single direction (no backtracking)
  - Avoids hotspots
  - Create flowcharts
- Number of readers <= Number of CPU cores
- Handle errors properly for Service Broker

Q & A

THANK YOU FOR COMING!

Feel free to contact me with questions/comments:

- Email: ed@leightondick.com
- Twitter: @eleightondick
- Blog: www.edleightondick.com

Please fill out your comment cards!