Service Broker: Async in the DB!

PASS SQLSATURDAY PHILADELPHIA | JUN 03 2017

Ryan Booz
@RyanBooz
ryan@SoftwareAndBooz.com

Agenda

- EnergyCAP Overview
- Discuss pain points Service Broker is solving for us
- Review the basics of Service Broker
- Use Cases
- Demo
- Common Problems
- Resources

The Backstory

My Recent Journey

- Attended SQL Server perf class @ CodeMash 2015
- My eyes were opened to how much I didn't know about SQL Server internals!
- I've been on a mission to overhaul our database, hosting environment and client-hosted guidance
- I'd say I'm 50% of the way there…
- … but loving every minute of it

EnergyCAP Ecosystem

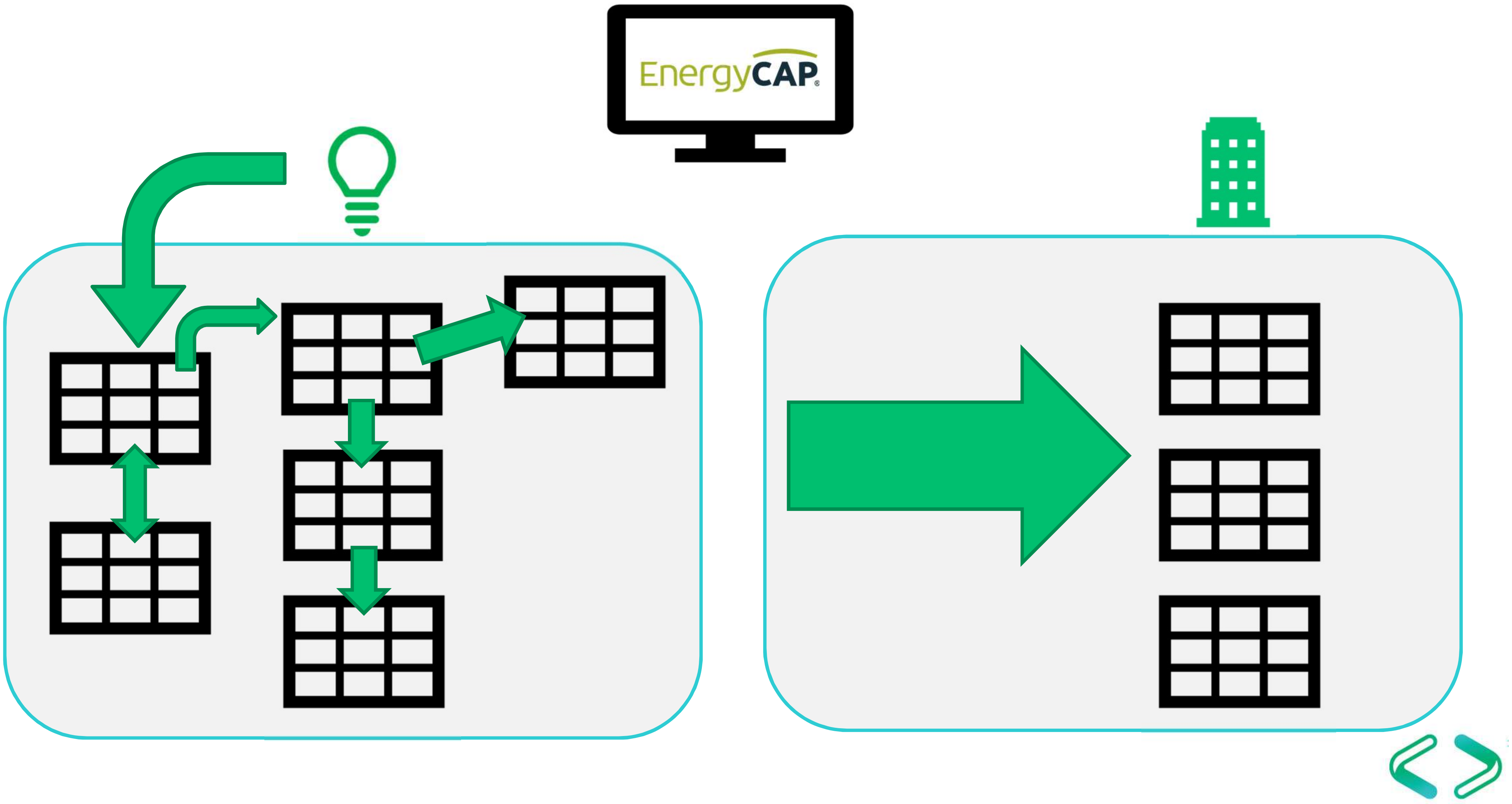
- Single DB accessed by three independent apps
  - Legacy desktop C++ client
  - (Almost) legacy Flash-based web app with .NET 4.5 web services
  - Modern Angular app backed by .NET Core web services
- SQL 2008R2, 2012 & 2014 currently in use
- SaaS offering with ~300 production databases (representing ~2000 clients)

EnergyCAP Ecosystem

- Client-hosted DBs are:
  - Often our largest databases
    - 3-10 million bills
    - 15-30 million bill detail lines
  - Almost always our limiting factor because of environment limitations

The Constant Thorn...

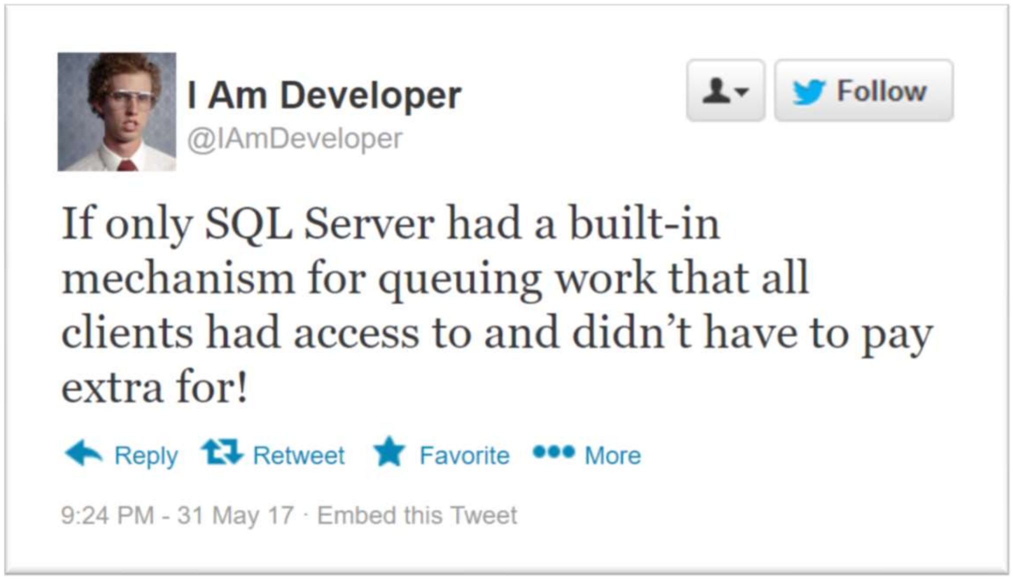
- As expected, business critical processing in SPROCs and Triggers
- However, the most repeated phrases in the last 15 years have been:
  - "Just add a flag..."
  - "Put it in the trigger..."
- A tangled web of triggers looking at flags, calling SPROCs, which call other SPROCs, which reset flags...


EnergyCAP

EnergyCAP Demo

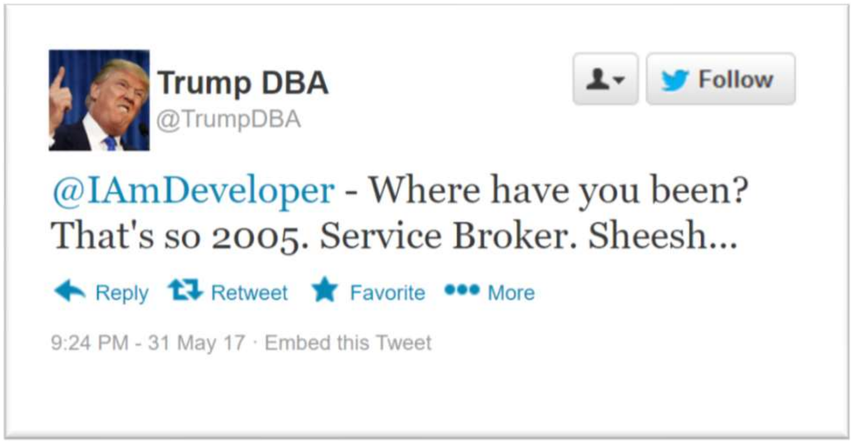
If Only...

- We explored external queuing and messaging systems many times
- Again, our self-hosted clients were always the limitation



If only SQL Server had a built-in mechanism for queuing work that all clients had access to and didn't have to pay extra for!

9:24 PM - 31 May 17 · Embed this Tweet



@IAmDeveloper - Where have you been? That's so 2005. Service Broker. Sheesh...

9:24 PM - 31 May 17 · Embed this Tweet

Enter Service Broker

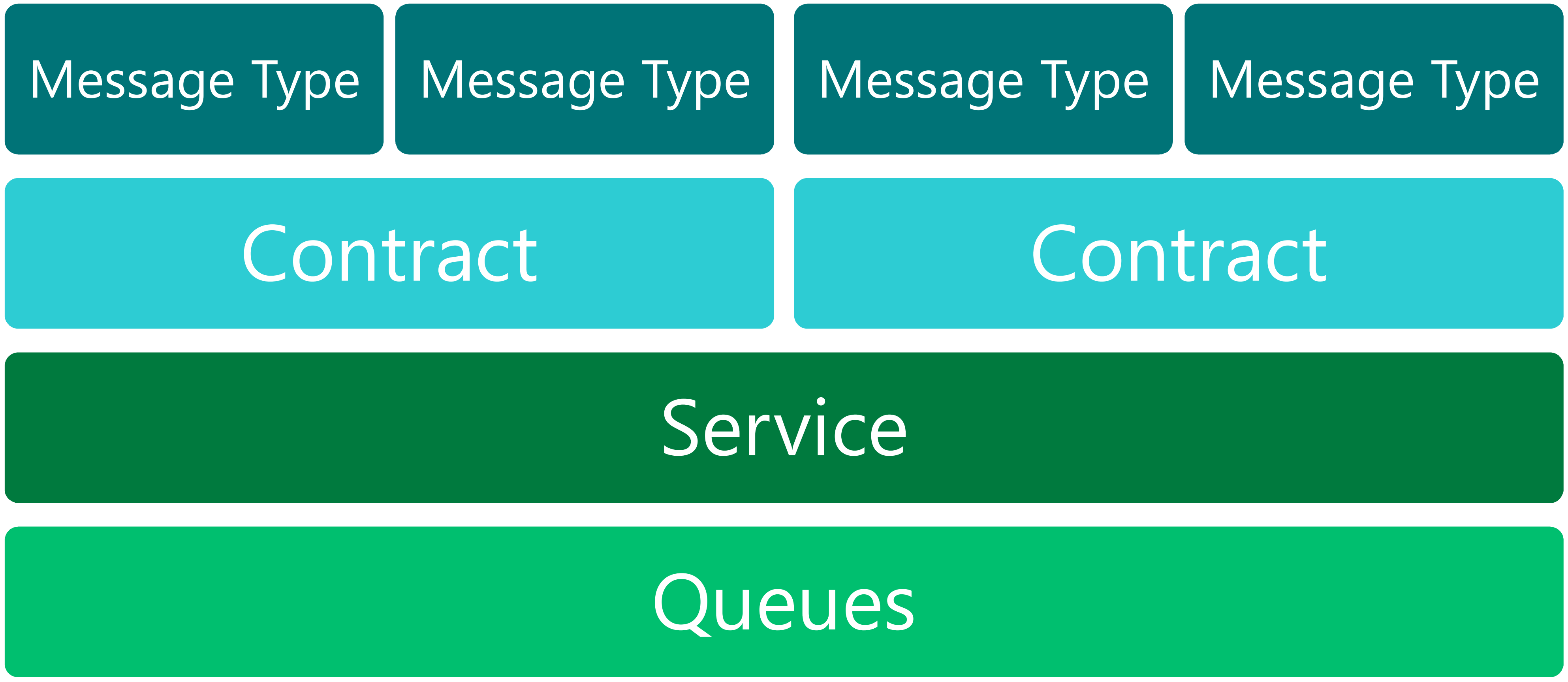
- A messaging system included with SQL Server 2005+, all versions
- Finally available in Azure with newly announced managed instances
- In DB, local instance, across instances, and remote messaging
- Reliable, transactional messaging

Service Broker:
The Basics

Conversations

- Service Broker apps are conversations
- There are (at least) two sides to the conversation
  - ***Not*** a fire-and-forget queuing system like MSMQ or RabbitMQ (this took some getting used to)
- The conversation must **always** be completed
- There is no out-of-the-box Pub/Sub capability

The Building Blocks



Queues

- Receives the messages
- Can be local (often) or remote (distributed messaging)
- Messages can be processed manually from external app or scheduled job
- Messages can also be processed automatically by a SPROC as they come in

Message Types

- Specifies the "name" of the message
- Can be validated (XML, Well Formed XML, Schema XML) or not

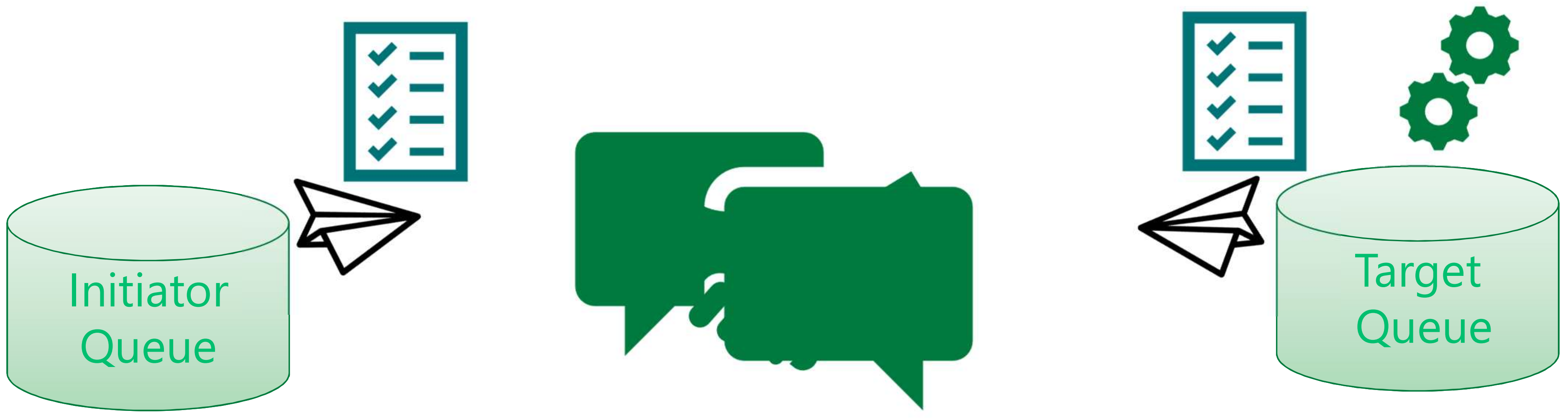
Contracts

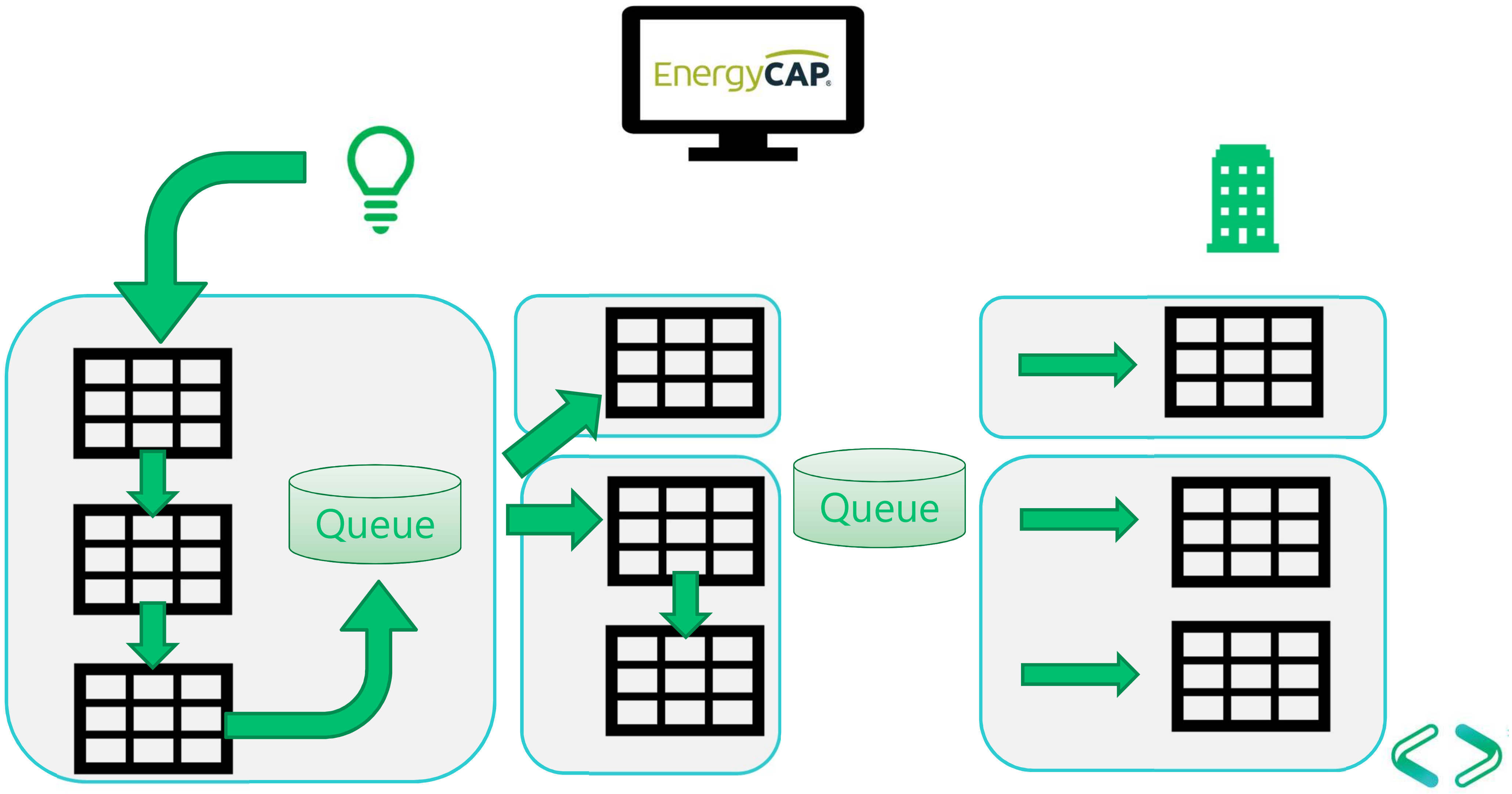
- Dictates what messages can be part of the conversation between services/queues
- Specifies which side of the conversation can send each type of message
- Multiple contracts can be added to a queue if necessary

Services

- The "traffic cop" of messages on a queue
- A service ensures that all messages in the conversation are supposed to be there

Basic Monologue Interaction




EnergyCAP
Queue
Queue

EnergyCAP Demo

Async Processing!

- Message processing (saving and retrieving) has been designed to be very lightweight and non-blocking
- The linear, synchronous processing in SPROCs and Triggers can now be ordered and adapted by adding messages to a queue
- Messages are processed by new threads, freeing the original transaction to return quickly

Common Uses

- Table logging (prior to Temporal Tables)
- Storing requests for long-running processes that can happen later (Reports)
- Committing information that starts other work out of process
- Shipping messages to another DB

Demo

Common Problems

- CLOSE THE CONVERSATION!
  - Otherwise, sys.conversation_enpoints will not clear out and bloat your database quickly!
- Enable broker any time a DB is restored
- Database ownership because SSB runs as the DB owner
- Still consider locking/blocking issues and timing (some rewrite of old processes is likely)

- When a SPROC is activated it will always attempt one last retrieve on a queue. Without proper error handline, you'll end up with messages in the SQL Server error log
- If you get an uncommittable transaction, the rollback can make it difficult to log the errors

Remember that triggers process in batches. If you need to do “per item” work, send needed rows as XML data and processes in a separate SPROC, not in the trigger!

Useful Links

- Remus Rusanu (http://rusanu.com/blog/)
- Dave Wentzel (http://www.davewentzel.com/taxonomy/term/343)
- Jonathan Kehayias (https://sqlperformance.com/2014/03/sql-performance/configuring-service-broker)