EventSourcing and CQRS

A DYNAMIC DUO THAT WILL SOLVE GLOBAL WARMING

(OR MAYBE SOMETHING A BIT MORE REALISTIC)

Prairie Dev Con 2022

Gold Sponsors

neo

SOLVERA
Part of Accenture

NORTHFIELD

online
business systems

Microsoft

LOTLINX®

Johnston
group

Community Supporters

iQmetrix®

RICHARDSON

Introductions

Dave White AKT, AKC, PSM, CSM, MVP Alumni

Principal Consultant – Depth Consulting

- In technology for over 20 years
- Cloud-native, DevOps, Architect, Developer
- Kanban/Agile Evangelist, Speaker
- WesternDevs member
- Alumni MS MVP (2014, 2015, 2016)

Linked in http://linkd.in/giMxuw

twitter @AgileRamblings

western devs

https://www.westerndevs.com

https://github.com/agileramblings

dave@depthconsulting.ca

Agenda

- Why are we here?
- EventSourcing
- CQRS
- Putting it all together
- Final Thoughts

Why are we here?

THE APPLICATION STORY WE’VE ALL (PROBABLY) HEARD BEFORE

Learn. Share. Laugh. Cry.

There once was an application…

- Started in about 2012
- ASP.NET MVC – .cshtml with JSON over HTTP RPC added along the way
- Single database
- Entity Framework
- 100 users or so
  - No meaningful customer integrations
  - Minimal reporting

7-8 years later…

- Still ASP.NET MVC – .cshtml with JSON over HTTP RPC added along the way
  - Many Telerik/KendoUI grids/components
- Single database
- Entity Framework still
- 3000 users that include people, devices and load from third-party vendor integrations
  - > $300 million dollars in annual revenue flow through this application
- BI ETLs run at night
- Large B2B EDI transactions run twice daily from major (read: $$$) customers

Sounds ok until…

- Users complain about application slowness *all the time*.
  - Some grids for LOB activities take minutes to load
  - Analysis suggests the database is the reason…
- Let's scale up the database hardware!
- Let's allow queries and other transactions to use uncommitted, in-progress transactional writes

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
```

- Let's spin up a read-only replica to try to take some of the load
- Call in the consultants to optimize data access

It didn’t work…

- Users continue to complain about application slowness *all the time.*
  - LOB activities still take minutes to load
  - Long-running nightly activities don’t finish by morning business hours
  - Analysis suggests the database is still the problem…
- The databases cost between $4k-6k per month in Azure
  - Always pressing up against the biggest memory SKU vms available in Azure
- Allowing transactions to interact creates data integrity problems
- No auditing. With 3000 users/agents and data integrity problems, this becomes a requirement. This may add some load to the database.
- Read-only replica isn’t having the expected impact
- There is nothing the consultants can do with the current architecture and EF

The problem wasn’t SQL Server.

We had to do something…

- Begin cleaving chunks out of the monolith
  - Don’t build another monolith
  - Don’t use the current monoliths database either
- Within a microservice, re-evaluate how to do data access
  - Ensure flexibility with regard to distributing workloads
  - Auditability is a requirement
  - This isn’t a talk about microservices
- Diverse consumers of business data need to be supported
  - REACT applications
  - ETL workloads
  - B2B service integrations
- Modern consumers have modern performance expectations

EventSourcing

DEMYSTIFYING AND MAKING IT LESS SCARY

Choosing EventSourcing as a persistence mechanism will have a profound impact on the overall implementation of your application.

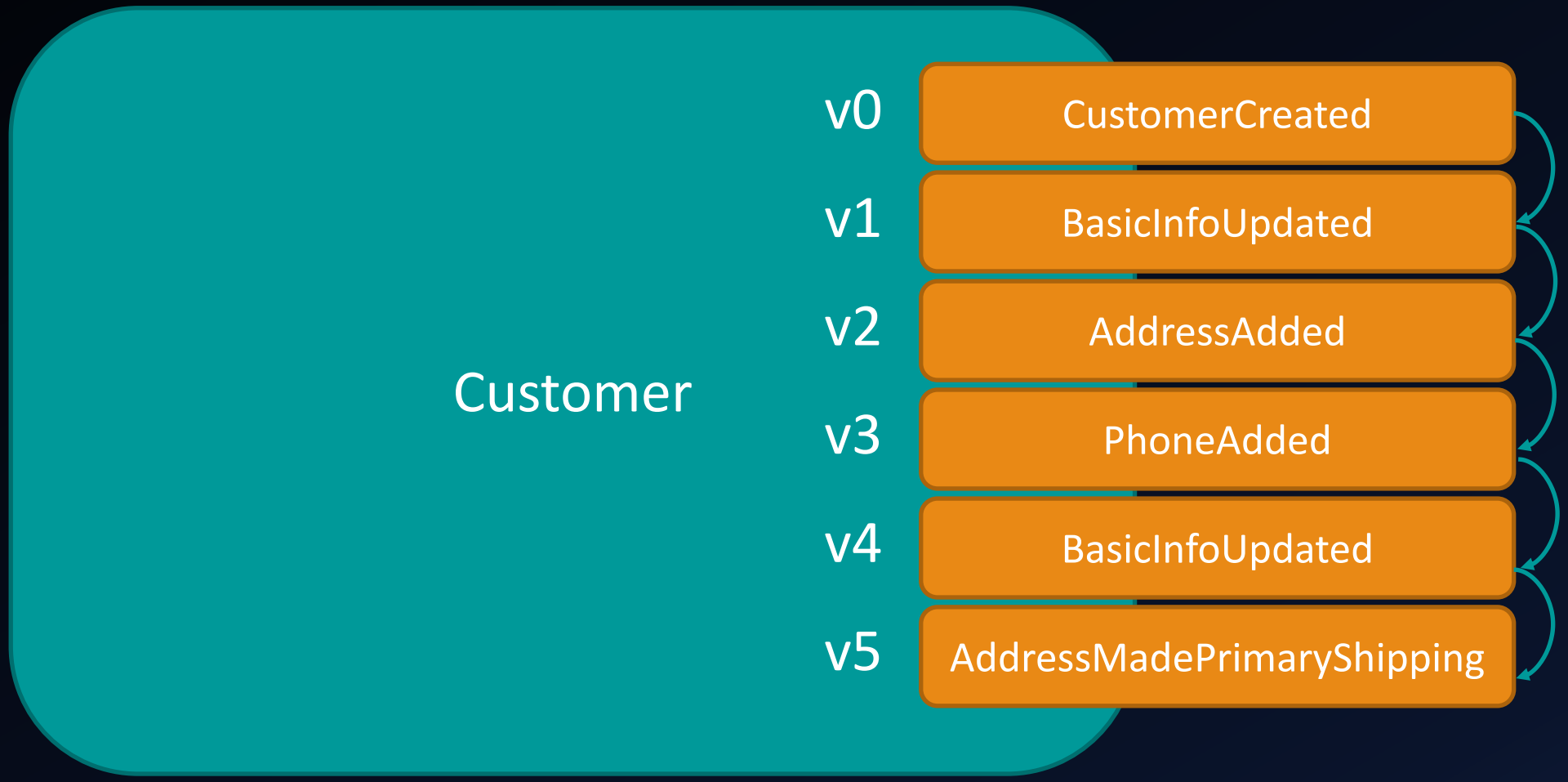
EventSourcing

- … is a different style for saving System of Record business data
- … aligns with alternate object modelling techniques (DDD)
- … provides an intuitive approach to modelling important business objects in code
- … is a bit more work
- … may support auditing requirements by default
- … won’t bite you

Technically speaking…

- Stream of events describing what happened to an object
- The stream is (should be) append-only. Events are immutable.
- Each event represents a new “version” of the object
  - Events happen at a point in time
- Re-hydrating the object means reading the stream and applying events one at a time in order

A Visual


Customer
v0
CustomerCreated
v1
BasicInfoUpdated
v2
AddressAdded
v3
PhoneAdded
v4
BasicInfoUpdated
v5
AddressMadePrimaryShipping

Event Store Records

SQLQuery2.sql - 127...ApiEvents (sa (68))* | SQLQuery1.sql - not connected

```sql
SELECT TOP (1000) [AggregateId]
      ,[Version]
      ,[AggregateType]
      ,[MessageId]
      ,[CorrelationId]
      ,[CausationId]
      ,[EventType]
      ,[EventData]
      ,[ReceivedOn]
  FROM [CustomerApiEvents].[dbo].[EventDescriptors]
  WHERE AggregateId = '0a362995-7d3b-4233-9656-94aaf6ff92db'
  ORDER BY Version asc
```

100 %

Results | Messages

	AggregateId	Version	AggregateType	MessageId	CorrelationId
1	0a362995-7d3b-4233-9656-94aaf6ff92db	0	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	266198AD-4E64-49D9-836E-B997E8816D84	correlationId
2	0a362995-7d3b-4233-9656-94aaf6ff92db	1	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	FDE051EB-5C83-4EAE-B074-2AE3FEA38E4E	correlationId
3	0a362995-7d3b-4233-9656-94aaf6ff92db	2	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	0CB175C7-0FDD-45B9-ABF1-AA33EA3920EF	correlationId
4	0a362995-7d3b-4233-9656-94aaf6ff92db	3	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	518BAC0B-AF15-467C-982B-1582DCC1EF5C	correlationId
5	0a362995-7d3b-4233-9656-94aaf6ff92db	4	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	32C5C206-C513-429B-A776-85545660FE51	correlationId
6	0a362995-7d3b-4233-9656-94aaf6ff92db	5	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	278EEBC7-8E16-45A7-9F22-DBF699624688	correlationId
7	0a362995-7d3b-4233-9656-94aaf6ff92db	6	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	9E2C3A7E-3266-4A5B-ADD5-FC251B1CD008	correlationId
8	0a362995-7d3b-4233-9656-94aaf6ff92db	7	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	6C7FB414-EF53-47A9-8482-2EFE6CD2E775	correlationId
9	0a362995-7d3b-4233-9656-94aaf6ff92db	8	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	65B3289E-5D3D-44F9-BFF4-2EE280523AC7	correlationId
10	0a362995-7d3b-4233-9656-94aaf6ff92db	9	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	DC5CFBAF-4284-4FAD-9148-B56F7D4A41A9	correlationId
11	0a362995-7d3b-4233-9656-94aaf6ff92db	10	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	8F84DDA7-1E70-4F6B-9D7B-A5B47054A4CB	correlationId
12	0a362995-7d3b-4233-9656-94aaf6ff92db	11	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	3EA462D9-31B4-4E54-A895-356237178117	correlationId
13	0a362995-7d3b-4233-9656-94aaf6ff92db	12	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	391B2D6C-D149-4093-919D-5E862C0F3EE3	correlationId
14	0a362995-7d3b-4233-9656-94aaf6ff92db	13	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	00A10C05-63D5-40C1-BE5C-CC437C4FBF06	correlationId
15	0a362995-7d3b-4233-9656-94aaf6ff92db	14	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	403301FB-5699-4FAA-9025-E80D4954FE6E	correlationId
16	0a362995-7d3b-4233-9656-94aaf6ff92db	15	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	55756CDD-0716-4CF6-8E53-0C3338A2A924	correlationId
17	0a362995-7d3b-4233-9656-94aaf6ff92db	16	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	F64235B5-74C2-48EB-9DCD-830C537644B7	correlationId
18	0a362995-7d3b-4233-9656-94aaf6ff92db	17	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	E95B2A68-F1C8-4E09-997B-096EDBCCA97E	correlationId
19	0a362995-7d3b-4233-9656-94aaf6ff92db	18	PRDC2022.Customer.Domain.Aggregates.Customer.Cus...	812C8750-988E-4007-9E64-E0763B016199	correlationId

vs Relational Records

SQLQuery4.sql - 127...piStaging (sa (70))* | SQLQuery2.sql - 127...ApiEvents (sa (68))*

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1) [AggregateId]
      ,[Version]
      ,[FirstName]
      ,[LastName]
      ,[MiddleName]
      ,[DateOfBirth]
      ,[Email]
      ,[Gender]
      ,[CreatedBy]
      ,[CreatedOn]
      ,[ModifiedBy]
      ,[LastModifiedOn]
  FROM [CustomerApiStaging].[dbo].[CustomerDetails]
  WHERE AggregateId = '0a362995-7d3b-4233-9656-94aaf6ff92db'
```

100 %

Results | Messages

	AggregateId	Version	FirstName	LastName	MiddleName	DateOfB
1	0a362995-7d3b-4233-9656-94aaf6ff92db	29	Marissa	Pearson	cosby	1979-10

SQLQuery5.sql - 127...piStaging (sa (74))* | SQLQuery4.sql - 127...piStaging (sa (70))* | SQLQuery2.sql - 127...ApiEver

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [PhoneId]
      ,[IsPrimary]
      ,[Number]
      ,[CountryCode]
      ,[PhoneType]
      ,[ParentId]
  FROM [CustomerApiStaging].[dbo].[Phones]
  WHERE ParentId = '0a362995-7d3b-4233-9656-94aaf6ff92db'
```

100 %

Results | Messages

	PhoneId	IsPrimary	Number	CountryCode	PhoneType	ParentId
1	D3F18523-5DD0-42D0-BC73-2606EA920EC1	0	447205029	4	occupy	0a362995-7d3b-4233-9656-94a
2	A1188DEE-4166-4D57-A819-2ADB020E4475	0	239902877	6	Leggings	0a362995-7d3b-4233-9656-94a
3	E5356942-2					
4	740BB68A-7					
5	D000979E-5					
6	30152DF8-3					
7	05F7947E-E					
8	A3AA9784-A					
9	DC3C787A-E					
10	70B91F10-1					
11	B93D489B-(					
12	0D5D7F9A-8					
13	DC549929-6					
14	6DC7DB65-2					
15	9FA23152-2					
16	334E6846-4					
17	16469B67-1					
18	650E71F4-F					
19	C47DF7FB-E					
20	3BD6E292-(					
21	D1D27E9A-(					
22	054860DF-9					

SQLQuery6.sql - 127...piStaging (sa (73))* | SQLQuery5.sql - 127...piStaging (sa (74))* | SQLQuery4.

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [AddressId]
      ,[IsPrimaryShipping]
      ,[IsPrimaryBilling]
      ,[Address1]
      ,[Address2]
      ,[City]
      ,[Region]
      ,[Country]
      ,[PostalCode]
      ,[ParentId]
  FROM [CustomerApiStaging].[dbo].[Addresses]
  WHERE ParentId = '0a362995-7d3b-4233-9656-94aaf6ff92db'
```

100 %

Results | Messages

	AddressId	IsPrimaryShipping	IsPrimaryBilling	Address1	A
1	331B8369-B39D-4B71-96F9-A8E5A50D11B4	1	0	8770 Paerdegat 12th Street	
2	0EA8C149-B193-4333-85A7-CA877F1B8C8E	0	0	7814 Pine Street	
3	1E2080B3-900A-4FCE-ABE9-EC02A0EB4E24	0	1	5627 Pembroke Street	

One Schema to Rule them All

- Adding data to an event stream is as simple as adding events types, stored in the stream
- Adding data to a relational model usually means new tables or discriminators (yuck!) to an ever widening table

A brief word about consistency…

- Immediately consistent (state) vs. eventually consistent
- Objects in memory are the only place where we can be certain of being **immediately** consistent
- Everything else should be considered **eventually** consistent
- Anywhere that data is <u>at rest</u> is **eventually** consistent
  - First one to save wins!
- This is why versioning of domain objects is valuable

(Domain) Event handling too!

- With event-sourced persistence, your application can respond to specific events when persisting new items to the stream
- Handlers are decoupled from workflows in code
  - Register event handlers and they are invoked by an in-process message bus
  - Projectors and Reactors are two broad categories of event handlers
    - Projectors emit projections (point in time picture of a business object)
    - Reactors initiate an out-of-process interaction (Post to API, place a message on a external message bus)

Pros/Cons

PROS

- Intuitive object modeling
- What to test (scenarios) are easier to think about
- Easy (read: none) schema management
- Built-in auditing/history retention
- Version control of business objects
- Supports more successful cloud-scale (distributed) workloads
- Write-heavy workloads required extra care

CONS

- More data (de-normalized) usually means more storage costs
- Acquisition costs (queries) from event stream are may be more expensive
  - Mitigated with snapshots/mementos
- Cannot easily query data directly anymore
- Objects need to support old events
- More complexity
- Different/novel

CQRS

SPREADING THE LOAD OVER SPACE AND TIME. AND… 42.

CQRS is not an architecture. It is a mindset.

CQRS

- … Command Query **Responsibility** Segregation
- … does *not* have to go hand in hand with EventSourcing
  - But they are a powerful combination
  - Always there are two…
- … provides an intuitive approach that guides fit-for-purpose persistence implementations to emerge
- … is a bit more work
- … won’t bite you

Commands

- Commands are a request to change the state of the world
- This usually means some sort of “write” activity

- e.g.
  - Please create a Customer record
  - Please add a phone number to a customer record

Query

- Queries are a request to see the state of the world
- This should not have any side-effects that change the state of the world
  - These side-effects are commands in disguise
- Queries (and responses) can (should?) be designed for specific consumers
  - JSON consumers
  - ETL integrations
  - Reporting tools

- e.g.
  - Get a customer record
  - Get a summary record of customer address/phone usage

Why should we keep these segregated?

- Ratio of workloads informs architectural design choices
  - Write workloads can be optimized/tuned
  - Query workloads can be optimized/tuned
  - May be deployed separately
- Can mitigate the “single persistence mechanism instance” risk
- Encourages fit-for-purpose query stores
  - NoSQL/Document databases
  - TimeSeries databases
  - Relational databases that are not attached the event store
  - Graph databases

Pros/Cons

PROS

- Encourages fit-for-purpose persistence mechanisms
- Separating command specific loads from query load allows provisioning “the right amount” of persistence compute
- Encourages architecturally consistent decisions
- Write-heavy workloads required extra care
- Query-heavy workloads can really benefit

CONS

- More data (de-normalized) usually means more storage costs
- More complexity
- Different/novel

Putting it all together

DID IT WORK THE WAY WE EXPECTED?

I have code!!!

Final Thoughts

SOME OF MINE, SOME OF YOURS

My thoughts

- I am a proponent of EventSourcing and CQRS and I am hoping that you will be too … someday.
- Someday will come if you aren’t intimidated by either of these techniques, which I am hoping you’ve taken away from today.
- It isn’t an all or nothing adoption
- Neither ES or CQRS will bite you
  - any more than any other code you may write
- Any novel technology comes with sharp edges
  - Practice. Be Safe. Learn. Collaborate.

Questions

- If ES or CQRS is still scary, why?
- Technical questions?
- Usage concerns?
- Adoption concerns?

Resources

- https://github.com/agileramblings
  - PRDC2022 – arriving soon
  - Battleship-CQRS
- Eric Evans – Domain Driven Design – Amazon.ca
- Vernon Vaughn – Implementing Domain Driven Design – Amazon.ca
- Greg Young
  - https://cqrs.files.wordpress.com/2010/11/cqrs_documents.pdf
  - https://www.youtube.com/watch?v=JHGkaShoyNs
  - https://www.eventstore.com/blog/author/greg-young

Thank you