ETL and Event Sourcing
Integration Architecture: Best Practice and Case Study
Marc Siegel - Panorama Education - Wed Feb 6 2019
ETL pipelines from external systems
ETL and Event Sourcing

Prerequisite knowledge

Familiarity with traditional ETL architectures:

Software systems that Extract data from external systems, Transform them, and Load the resulting data sets into internal systems, most often relational databases

Dissatisfaction with traditional ETL architectures / curiosity to learn about and consider an alternative architecture
ETL and Event Sourcing

What you’ll learn

How Event Sourcing can be applied to ETL

How Determinism can be a property of a system

Value of treating the Past as First Class
What is ETL?
ETL

In a nutshell
ETL

In a nutshell

External System
ETL

In a nutshell

External System

Traditional ETL Process

Extract
ETL

In a nutshell
ETL

In a nutshell
ETL

In a nutshell
ETL

In a nutshell

Q: What is the System of Record? What is the Source of Truth?
ETL

In a nutshell

System of Record

The authoritative data source for a given data element or piece of information (1)
ETL

In a nutshell

**Source of Truth**

A trusted data source that gives a *complete picture* of the data object as a whole (2)
ETL

In a nutshell

External System

Traditional ETL Process
Extract → Transform → Load

Internal Database
ETL Challenges

- Operational
- Domain Modelling
- Selective Attention
ETL Challenges

Operational
Must rerun long ETL job to test edge case

Domain Modelling
Missing Interests:
- Decoupling

Selective Attention
ETL Challenges

Operational

Must rerun long ETL job to test edge case

Running ETL job can overwrite history

Domain Modelling

Missing Interests:
- Decoupling
- Determinism

Selective Attention
## ETL Challenges

### Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ETL Challenges

- **Operational**
  Must create *one true schema* to load into

- **Domain Modelling**
  Missing Interests:
  - Decoupling (of each interpretation)

- **Selective Attention**
ETL Challenges

Operational

Must create one true schema to load into

Tend toward lowest common denominator
OR superset of all external model features

Domain Modelling

Selective Attention

Missing Interests:

- Decoupling (of each interpretation)
- Modeling State Explicitly
## ETL Challenges

### Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Modeling State Explicitly</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ETL Challenges

Operational

Domain Modelling

Selective Attention

From Psychology: *the act of focusing on a particular object while ignoring irrelevant information*

→ Can’t re-interpret past extracts

**Missing Interests:**

- Past as First Class
How many passes did the team in white make?
ETL Challenges

Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ETL Advantage

Not just problems. Positive trade-offs of ETL?

- Low Costs: Training, framing, explaining
  - **Training**: Low cost to train new engineers in ETL concepts
  - **Framing**: No requirement for explicit domain modeling
  - **Explaining**: Intuitive to explain to non-engineers
## ETL Challenges

### Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Low Cost</strong></td>
<td><strong>X</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is ELT?
ETL

In a nutshell

Traditional ETL Process

Extract ➔ Transform ➔ Load

External System ➔ Internal Database
ETL and ELT

Traditional ETL Process

Extract  Transform  Load

External System  Internal Database
ETL and ELT

Traditional ETL Process
- Extract
- Transform
- Load

EL Process
- Extract
- Load

 External System

Internal Database
ETL and ELT

Traditional ETL Process

Extract → Transform → Load

Internal Database

EL Process

Extract → Load

Data Lake or Blob or File Store
ETL and ELT

Traditional ETL Process

Extract → Transform → Load

Internal Database

EL Process

Extract → Load

Data Lake or Blob or File Store

T Process
Do anything here! Many vendors offering various solutions.
ETL and ELT

Traditional ETL Process:
- Extract
- Transform
- Load

Internal Database

External System

EL Process:
- Extract
- Load

Data Lake or Blob or File Store

(T Process(es)
Do anything here! Many vendors offering various solutions.

oreillysacon.com/ny
#OReillySACon
# ETL and ELT

## Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## ETL and ELT

### Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Determinism</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ETL and ELT

Traditional ETL Process:
- Extract
- Transform
- Load

Internal Database

EL Process:
- Extract
- Load

Data Lake or Blob or File Store

T Process(es)
Do anything here! Many vendors offering various solutions.

External System
## ETL and ELT

### Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Modeling State Explicitly</strong></td>
<td></td>
<td><strong>blue</strong></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td><strong>blue</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# ETL and ELT

## Interests and Positions

<table>
<thead>
<tr>
<th>Interest</th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# ETL and ELT

## Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is Event Sourcing?
ETL

In a nutshell
ETL and ELT

Traditional ETL Process
Extract → Transform → Load

External System

EL Process
Extract → Load

Data Lake or Blob or File Store

T Process(es)
Do anything here! Many vendors offering various solutions.
ETL and Event Sourcing

Traditional ETL Process
- Extract
- Transform
- Load

EL Process
- Ex
- Lo

Internal Database

External System
ETL and Event Sourcing

Traditional ETL Process
- Extract
- Transform
- Load

Internal Database

External System

EL Process
- Extract
- Load

Immutable & Sequential Store
ETL and Event Sourcing

Traditional ETL Process

Extract ➔ Transform ➔ Load

Internal Database

External System

EL Process

Ex ➔ Lo ➔ Immutable & Sequential Store

TeTL Process
ETL and Event Sourcing

Traditional ETL Process

Extract → Transform → Load

Internal Database

External System

EL Process

Ex → Lo → Immutable & Sequential Store

TeTL Process

Tr → Domain Events

O’REILLY
Software Architecture

oreillysacon.com/ny
#OReillySACon
ETL and Event Sourcing

Traditional ETL Process

Extract ➔ Transform ➔ Load

Internal Database

External System

EL Process

Ex ➔ Lo

Immutable & Sequential Store

TeTL Process

Tr ➔ Domain Events ➔ Tr ➔ Lo

O’REILLY®
Software Architecture

oreillysaccon.com/ny
#OReillySACon
ETL and Event Sourcing

**Traditional ETL Process**
- Extract
- Transform
- Load

**Internal Database**

**External System**

**EL Process**
- Extract (Ex)
- Load (Lo)
- Immutable & Sequential Store

**TeTL Process**
- Transform (Tr)
- Domain Events
- Load (Lo)

**Read Model**
ETL and Event Sourcing

Traditional ETL Process:
- Extract
- Transform
- Load

Internal Database

External System

EL Process:
- Extract (Ex)
- Load (Lo)

Immutable & Sequential Store

TeTL Process(es):
- Transform (Tr)
- Domain Events
- Transform (Tr)
- Load (Lo)

Read Model(s)
ETL and Event Sourcing

Traditional ETL Process

Extract → Transform → Load → Internal Database

External System

EL Process

Ex → Lo

Immutable & Sequential Store

TeTL Process(es)

Tr → Domain Events → Tr → Lo

Read Model(s)

1) **Decouple** extractions

2) **Source of Truth**: the extracts

3) **Deterministic** transform: to events + to model

regular expression mnemonic: from /ETL/ to /E\{1\}T*L*/ ← Extract once, Transform & Load Infinitely
# ETL, ELT, and Event Sourcing

## Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## ETL, ELT, and Event Sourcing

### Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Determinism</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# ETL, ELT, and Event Sourcing

**Interests and Positions**

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Modeling State Explicitly</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ETL and Event Sourcing

Traditional ETL Process

1) Decouple extractions
2) Source of Truth: the extracts
3) Deterministic transform: to events + to model

regular expression mnemonic: from /ETL/ to /E{1}T*L*/ ← Extract once, Transform & Load Infinitely
# ETL, ELT, and Event Sourcing

## Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Event Sourcing Challenge

Not just advantages. Negative trade-offs of ES?

- High Costs: Training, framing, explaining
  - **Training**: Higher cost to train new engineers in ES concepts
  - **Framing**: Requirement for (lots of) explicit domain modeling
  - **Explaining**: Not necessarily intuitive to explain to non-engineers
ETL, ELT, and Event Sourcing

Interests and Positions

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
<th>Event Sourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determinism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling State Explicitly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Past as First Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PROBLEMS

SOLUTIONS
How does Event Sourcing work?
Event Sourcing Basics

Events

GradeCreated
student_id: 123
course_id: abc
grade: B+

GradeUpdated
student_id: 123
course_id: abc
grade: C

GradeUpdated
student_id: 123
course_id: abc
grade: A-
Event Sourcing Basics

Events

State transitions are an important part of our problem space and should be modeled within our domain.
Event Sourcing Basics

Events

*State transitions are an important part of our problem space and should be modeled within our domain.*

*Event Sourcing says all state is transient and you only store facts.*
Event Sourcing Basics

Events

State transitions are an important part of our problem space and should be modeled within our domain.

Event Sourcing says all state is transient and you only store facts.

**Event**: something that happened in the past; a fact; a state transition.
Event Sourcing Basics

Events

- **GradeCreated**
  - student_id: 123
  - course_id: abc
  - grade: B+

- **GradeUpdated**
  - student_id: 123
  - course_id: abc
  - grade: C

- **GradeUpdated**
  - student_id: 123
  - course_id: abc
  - grade: A-
Event Sourcing Basics

Events

- GradeCreated
  - student_id: 123
  - course_id: abc
  - grade: B+

- GradeUpdated
  - student_id: 123
  - course_id: abc
  - grade: C

- GradeUpdated
  - student_id: 123
  - course_id: abc
  - grade: A-

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>B+</td>
</tr>
</tbody>
</table>
Event Sourcing Basics

Events

GradeCreated
student_id: 123
course_id: abc
grade: B+

GradeUpdated
student_id: 123
course_id: abc
grade: C

GradeUpdated
student_id: 123
course_id: abc
grade: A-

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>C</td>
</tr>
</tbody>
</table>
Event Sourcing Basics

Events

GradeCreated
student_id: 123
course_id: abc
grade: B+

GradeUpdated
student_id: 123
course_id: abc
grade: C

GradeUpdated
student_id: 123
course_id: abc
grade: A-

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>
Event Sourcing Basics

Read Models

*Event Sourcing takes the term Read Model from CQRS.*
Event Sourcing Basics

Read Models

*Event Sourcing takes the term Read Model from CQRS.*

*A Read Model is an interpretation of a sequence of events, that is optimized for answering a given set of queries (reads).*
Event Sourcing Basics

Read Models

*Event Sourcing takes the term Read Model from CQRS.*

A Read Model is an interpretation of a sequence of events, that is optimized for answering a given set of queries (reads).

*Read Models: are independent representations of state that we deterministically regenerate from events using projections.*
**Event Sourcing Basics**

### Events

**GradeCreated**
- student_id: 123
- course_id: abc
- grade: B+

**GradeUpdated**
- student_id: 123
- course_id: abc
- grade: C

**GradeUpdated**
- student_id: 123
- course_id: abc
- grade: A-

### Projections

```python
def f(state, event)
    state.where(
        student_id: event.student_id,
        course_id: event.course_id
    ).update(grade: event.grade)
end
```

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>
Event Sourcing Basics

Projections

*When we talk about Event Sourcing, current state is a left-fold of previous behaviors.*
Event Sourcing Basics

Projections

When we talk about Event Sourcing, current state is a left-fold of previous behaviors.

We play back a stream of events, applying a function

\[ f(\text{state}_n, \text{event}_n) \rightarrow \text{state}_{n+1} \]
Event Sourcing Basics

Projections

*When we talk about Event Sourcing, current state is a left-fold of previous behaviors.*

*We play back a stream of events, applying a function* \( f \left( \text{state}_n, \text{event}_n \right) \rightarrow \text{state}_{n+1} \)

**Projection**: a function through which we apply events in sequence to deterministically derive the state of our application.
Event Sourcing Basics

Events

GradeCreated
student_id: 123
course_id: abc
grade: B+

GradeUpdated
student_id: 123
course_id: abc
grade: C

GradeUpdated
student_id: 123
course_id: abc
grade: A-

Projections

def f(state, event)
    state.where(
        student_id: event.student_id,
        course_id: event.course_id
    ).update(grade: event.grade)
end

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>
Event Sourcing Basics

Review

**Event**: something that happened in the past; a fact; a state transition.

**Projection**: a function through which we apply events in sequence to deterministically derive the state of our application.

**Read Models**: are independent representations of state that we deterministically regenerate from events using projections.
Event Sourcing Basics

Events

**GradeCreated**
- student_id: 123
- course_id: abc
- grade: B+

**GradeUpdated**
- student_id: 123
- course_id: abc
- grade: C

**GradeUpdated**
- student_id: 123
- course_id: abc
- grade: A-

Projections

```python
def f(state, event)
    state.where(
        student_id: event.student_id,
        course_id: event.course_id
    ).update(grade: event.grade)
end
```

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>
95% of the cost of event sourcing projects is explaining event sourcing
Applying Event Sourcing to ETL

Q: How to we get from ETL to explicitly modeled Domain Events?
Applying Event Sourcing to ETL

Q: How do we get from ETL to explicitly modeled Domain Events?
Applying Event Sourcing to ETL

Q: How to we get from ETL to explicitly modeled Domain Events?

A: Build an **Observational** Event Sourced system
Applying Event Sourcing to ETL

**Observations**

- Oct 9, 2018
  - Biology: B-

- Oct 10, 2018
  - History: C

- Oct 11, 2018
  - Spanish: A-

- Oct 12, 2018
  - Biology: B+

**Domain Events**

- GradeUpdated
  - student_id: 123
  - course_id: abc
  - grade: A-

**Read Models**

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>
Applying Event Sourcing to ETL

Observational

When capturing observations of external systems using Event Sourcing, the events in our domain are the observations we capture.
Applying Event Sourcing to ETL

Observational

When capturing observations of external systems using Event Sourcing, the events in our domain are the observations we capture.

Transforming a sequence of observations into explicitly modeled domain events is the first projection.
Applying Event Sourcing to ETL

Observational

When capturing observations of external systems using Event Sourcing, the events in our domain are the observations we capture.

Transforming a sequence of observations into explicitly modeled domain events is the first projection.

**Observational**: an Event Sourced system where the event history is of captured observations, and all state is derived from them.
Applying Event Sourcing to ETL

Observations

Domain Events

Read Models

GradeUpdated
student_id: 123
course_id: abc
grade: A-

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>
Applying Event Sourcing to ETL

Observations

Domain Events

GradeUpdated
student_id: 123
course_id: abc
grade: A-

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>

Immutable & Sequential Store
Applying Event Sourcing to ETL

Observations

<table>
<thead>
<tr>
<th>Date</th>
<th>Subject</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 9, 2018</td>
<td>Biology</td>
<td>B-</td>
</tr>
<tr>
<td>Oct 10, 2018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct 11, 2018</td>
<td>Biology</td>
<td>B+</td>
</tr>
<tr>
<td>Oct 12, 2018</td>
<td>History</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Spanish</td>
<td>A-</td>
</tr>
</tbody>
</table>

Domain Events

GradeUpdated
student_id: 123
course_id: abc
grade: A-

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>

TeTL Process(es)

Immutable & Sequential Store

Tr

Domain Events
Applying Event Sourcing to ETL

Observations

Oct 9, 2018
Oct 10, 2018
Oct 11, 2018
Oct 12, 2018

Biology
History
Spanish

B+
C
A-

GradeUpdated
student_id: 123
course_id: abc
grade: A-

Domain Events

Read Models

<table>
<thead>
<tr>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>abc</td>
<td>A-</td>
</tr>
</tbody>
</table>

Immutable & Sequential Store

TeTL Process(es)

Read Model(s)
Case study: Event Sourcing ETL

Teachers & Administrators

- Updates a grade
- Adds student to intervention
- Records behavior incident

Teacher interface into SIS

Database of SIS

Panorama Student Success

Exporter

Export History

- Oct 9, 2018
- Oct 10, 2018
- Oct 11, 2018
- Oct 12, 2018

Biology: B-
History: C
Spanish: A-

Takes a snapshot of the SIS's current state
Case study: Event Sourcing ETL

observation events

projection

 GradeUpdated
 student_id: 1
date: Oct 11
course: Biology
grade: B-

 GradeUpdated
 student_id: 1
date: Oct 12
course: Biology
grade: B+
Case study: Event Sourcing ETL

```
GradeUpdated
  student_id: 1
  date: Oct 11
  course: Biology
  grade: B-

GradeUpdated
  student_id: 1
  date: Oct 12
  course: Biology
  grade: B+
```

projection

InProgressGrades

read models

domain events
Case study: Event Sourcing ETL

InProgressGrades

queried

What are Marc’s grades?

<table>
<thead>
<tr>
<th>Term</th>
<th>Course</th>
<th>Teacher</th>
<th>Changed</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-19</td>
<td>ETL and Event Sourcing</td>
<td>Tim O’Reilly</td>
<td>20 days ago</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>
Case study: Event Sourcing ETL

Past as First Class

First

Later interpretation
Case study: Event Sourcing ETL
Past as First Class

<table>
<thead>
<tr>
<th>Term</th>
<th>Course</th>
<th>Teacher</th>
<th>Changed</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-19</td>
<td>ETL and Event Sourcing</td>
<td>Tim O’Reilly</td>
<td>20 days ago</td>
<td>B</td>
<td>A</td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

Later interpretation
Case study: Event Sourcing ETL
Past as First Class

<table>
<thead>
<tr>
<th>Term</th>
<th>Course</th>
<th>Teacher</th>
<th>Changed</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-19</td>
<td>ETL and Event Sourcing</td>
<td>Tim O'Reilly</td>
<td>20 days ago</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

Later interpretation
Case study: Event Sourcing ETL

Determinism
Case study: Event Sourcing ETL

Determinism

- Read Models regenerated **nightly** from source of truth
  - Given the same history, we regenerate the same Read Models
Case study: Event Sourcing ETL

Determinism

- Read Models regenerated nightly from source of truth
  - Given the same history, we regenerate the same Read Models

- On-demand Read Model Comparison tool
  - Ensure no Read Model changes across larger code refactors
Case study: Event Sourcing ETL

Determinism

Read Model Comparison - Before and After Regeneration

Read Model DB

Clone Read Model

regen

batch_Before

Regenerations Run

Same DB, but later.

Clone Read Model Again

batch_After
Case study: Event Sourcing ETL

**Determinism**

Read Model Comparison - Before and After Regeneration

Read Model DB → Regenerations Run → Same DB, but later.

- **Unchanged Models:**
  - incidents_1, gpa_1, student_search_1, ...

- **Missing models from left read model:**
  - NONE

- **Missing models from right read model:**
  - trends_2, support_plans_1

--- Differences ---

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>year_1</td>
<td>~1</td>
</tr>
<tr>
<td>attendance_1</td>
<td>~3</td>
</tr>
<tr>
<td>indicators_2</td>
<td>~1</td>
</tr>
</tbody>
</table>
Case study: Event Sourcing ETL

Determinism

Read Model Comparison - Before and After Regeneration

Unchanged Models:
incidents_1, gpa_1, student_search_1, ...

Missing models from left read
NONE

Missing models from right read
trends_2, support_plans_1

------------------- Differences -------------------
Model  :   Number  
year_1  :   ~1   
attendance_1  :   ~3   
indicators_2  :   ~1   

3c3
< {:client_id="5d535703-8131-4878-b74d-a262a8ec314c", :date=>Mon, 27 Aug 2018, 
  :attendance_code=>"GOOSE GOOSE GOOSE Absent", :is_present=>false, :is_derived=>false}

6c6
> {:client_id="5d535703-8131-4878-b74d-a262a8ec314c", :date=>Mon, 27 Aug 2018, 
  :attendance_code=>"MOOSE MOOSE MOOSE Absent", :is_present=>false, :is_derived=>false}

---

---

> {:client_id="5d535703-8131-4878-b74d-a262a8ec314c", :date=>Mon, 27 Aug 2018, 
  :attendance_code=>"Present AND AWESOME", :is_present=>true, :is_derived=>false}

---

> {:client_id="5d535703-8131-4878-b74d-a262a8ec314c", :date=>Mon, 27 Aug 2018, 
  :attendance_code=>"Present AND NOISY", :is_present=>true, :is_derived=>false}
Case study: Event Sourcing ETL

Trade-off: Investment in Training
Case study: Event Sourcing ETL

Trade-off: Investment in Training

- 5 x 1 hr training videos + 1 hr discussions = 10 hrs
Case study: Event Sourcing ETL

Trade-off: Investment in Training

- 5 x 1 hr training videos + 1 hr discussions = 10 hrs
- Gentle ramp up w/ pairing and joint designs (weeks)
Case study: Event Sourcing ETL

Trade-off: Investment in Training

- 5 x 1 hr training videos + 1 hr discussions = 10 hrs
- Gentle ramp up w/ pairing and joint designs (weeks)
- Set expectation that architecture will feel different
Lessons Learned

At the two year mark

- Lessons learned: Thinnest extractions possible
- Lessons learned: Extracted files as Source of Truth
- Lessons learned: Many iterations on transformations
- Lessons learned: Why TL must be fast and run often
Lessons Learned

At the two year mark

Lessons learned: Thinnest extractions possible

My first version of converting [one type of] XML to CSV was silently dropping rows, and would have lost all that data if not for the ability to replace from original extract.
Lessons Learned

At the two year mark

Lessons learned: Extracted files as Source of Truth

*Real world example of changing incorrect foreign key reference (which had been nearly all overlapping previously).*
Lessons Learned
At the two year mark

Lessons learned: Many iterations on interpretations

Very natural to handle the changes, big and small, that appear in the format and content of the data we have extracted. Also, new features sometimes mean new or changed interpretations.
Lessons Learned

At the two year mark

Lessons learned: Why TL must be fast and run often

Consider the “nightly restores from backups” to prove that you can actually restore from backups. This practice exists in our application rather than our tools. If regeneration ever gets too slow to complete overnight, we could lose this.
Summary and Review

What we covered

- How Event Sourcing can be applied to ETL
- How Determinism can be a property of a system
- Value of treating the Past as First Class
Learn More

Resources

- DDD, CQRS, and Event Sourcing videos by Greg Young
- CQRS documentation site by Edument AB
- Domain Driven Design book by Eric Evans

Keep in touch!

- twitter: @ms_ati
- email: msiegel@panoramaed.com