DROPSOURCE

Event Sourcing at Dropsource with Node.js

By: Fernando Trigoso

May 10, 2016

What is Event Sourcing?

When there was a new event

- New piece of paper
- Date it
- Write down the event
- Append it at the end of the record

Characteristics of storing data this way

- Time is built-in
- Changes are incremental and immutable
- Records could grow very large
- Could go back and review employee history, changes, hidden trends, etc.
- Future proof

Early computers

- Limited memory
- Not very fast

Computer programmers back then

- Stored only current state
- Preemptively designed entities and relationships

name	phone
peter	~~555-4444~~

name	phone
peter	333-8888

Not future proof

- What if people start having multiple phone numbers?
- Changing initial design was difficult

It is still difficult

- Sql databases
- Document-oriented databases
- Has your data migration ever gone wrong?

Other smells of only storing current state

- Have to attach analytics tools to our applications
- Effectively dating records
- Big upfront design

It is 2016

Modern computer systems

- Storage is cheap and abundant
- Computers are super fast

The time has come for:

- Event Sourcing
- CQRS
- Fact-based system

	T1	T2	T3	T4	T5
peter	...	phone: 555-4444	...	....	phone: 333-8888

- Snapshot of database at point-in-time
- Changes since point-in-time

	T1	T2	T3	T4	T5
peter	…	phone: 555-4444	…	….	phone: 333-8888

- Previous events that need to be reversed
- It is future proof

Who deals with changes in data?

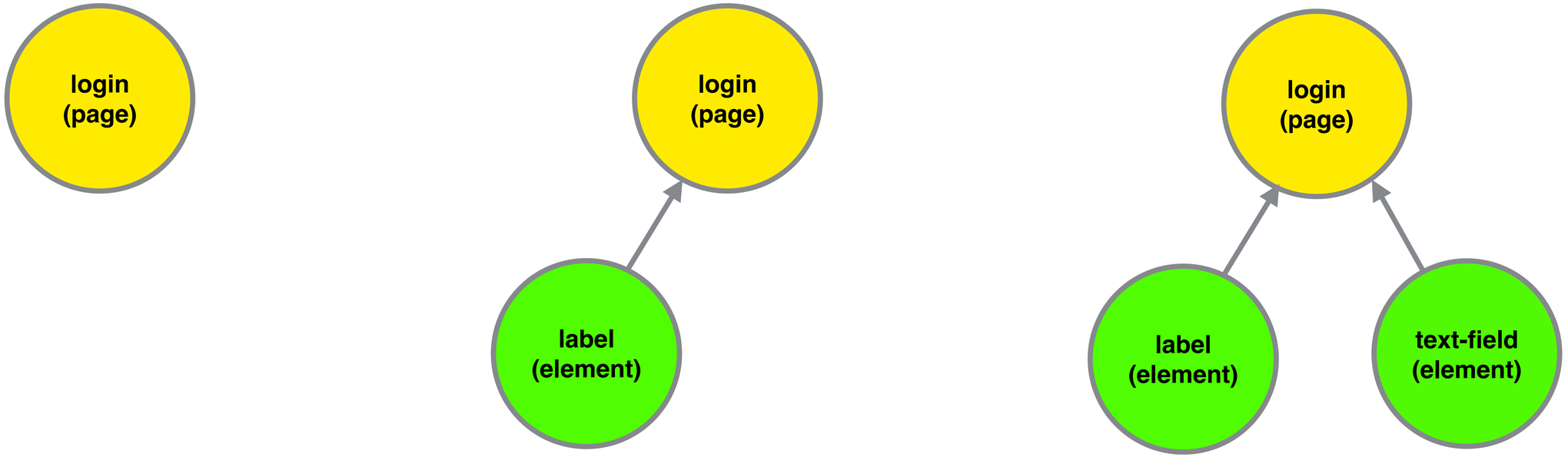
- The application

Event Sourcing at Dropsource

- Only workbench uses ES
- We store events in Couchbase
- Every Dropsource project has its own event stream
- We create two projections off the event stream

T1	T2	T3

`{`
`  eventType: ‘pageCreated’,`
`  data:`
`  {`
`  id: ‘login’`
`  }`
`}`
	`{`
	`  eventType: ‘elementCreated’,`
	`  data:`
	`  {`
	`  id: ‘label’,`
	`  parent: ‘login’`
	`  }`
	`}`
		`{`
		`  eventType: ‘elementCreated’,`
		`  data:`
		`  {`
		`  id: ‘text-field’,`
		`  parent: ‘login’`
		`  }`
		`}`



- Events are immutable
- We cache projections

Advantages of using Javascript

- Workbench and Node API create events
- Both run javascript
- Reuse code
  - Creates events
  - Generates projections

More advantages of using Javascript

- Better UX
  - Client doesn’t have to wait for new projections
  - Optimistic save

We send all events to the Workbench (browser)

- Enables better features, like team collaboration
- No need to push large new projections to the browser
- Need to manage initial load time

Other benefits of using ES

- Reversing events: undo/redo
- Versioning of projects
- Analytics, user behavior
- Fits nicely with React and Flux

ES-related technologies I’m interested in

- Datomic
  - Transactional, distributed database, built-in audit, time-based
  - http://www.datomic.com/
- Gorilla
  - Facebook’s in-memory time series database
  - http://www.vldb.org/pvldb/vol8/p1816-teller.pdf
- Event Store
  - Time series database with event processing in Javascript
  - https://geteventstore.com/

Thank you!