

A scenic landscape featuring a dark road curving through a field of tall, golden grass. In the background, there are rolling hills under a dramatic sky filled with large, textured clouds. The sun is low on the horizon, casting a warm, golden glow over the scene.

HOW WE USED
REACT
TO TAME THE SINGLE-PAGE BEAST

@EWNDL

ERIK WENDEL

JAVASCRIPTER & CONSULTANT

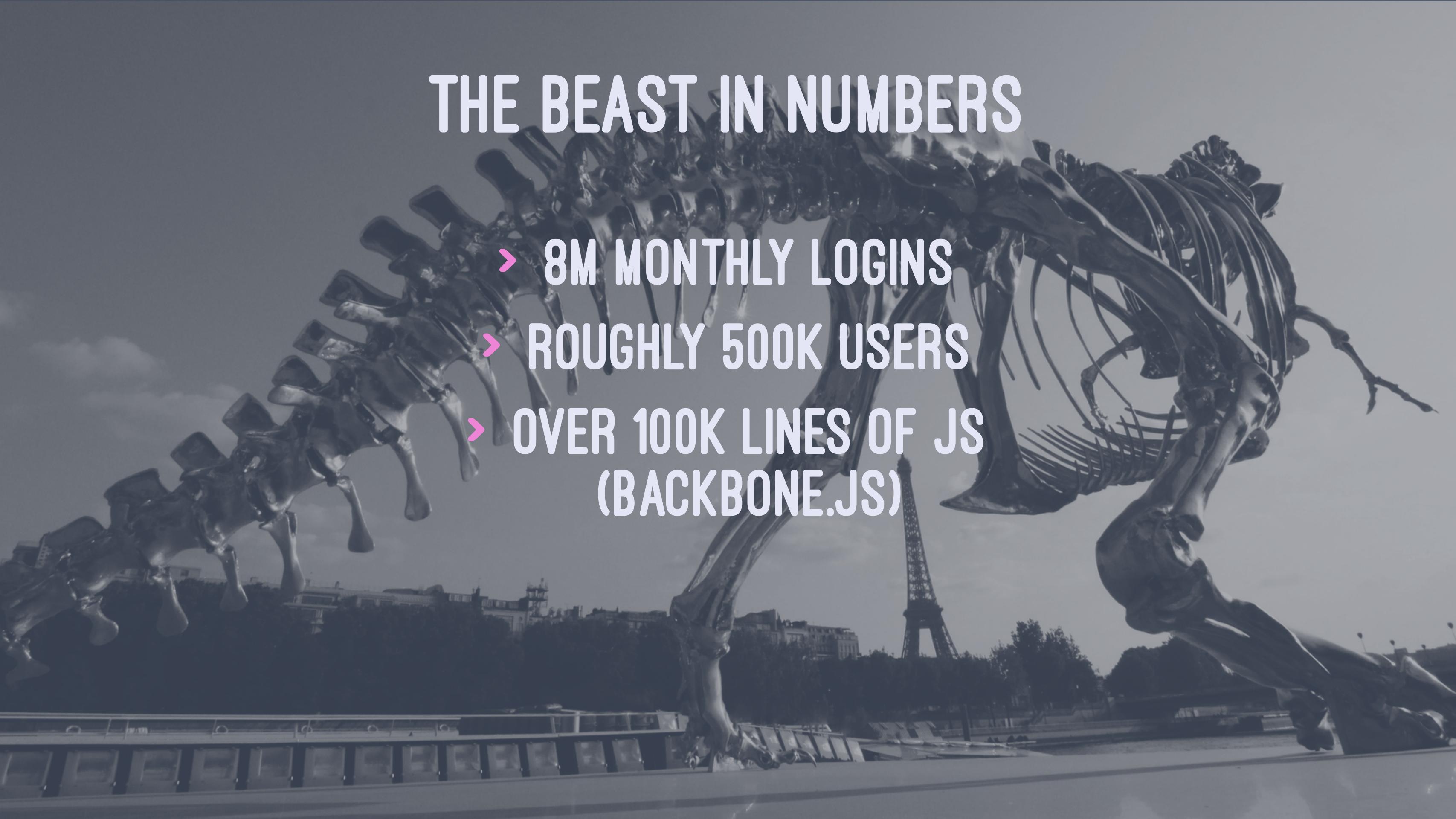
BEKK CONSULTING. OSLO. NORWAY

PRACTICE LEAD. WEB TECHNOLOGIES

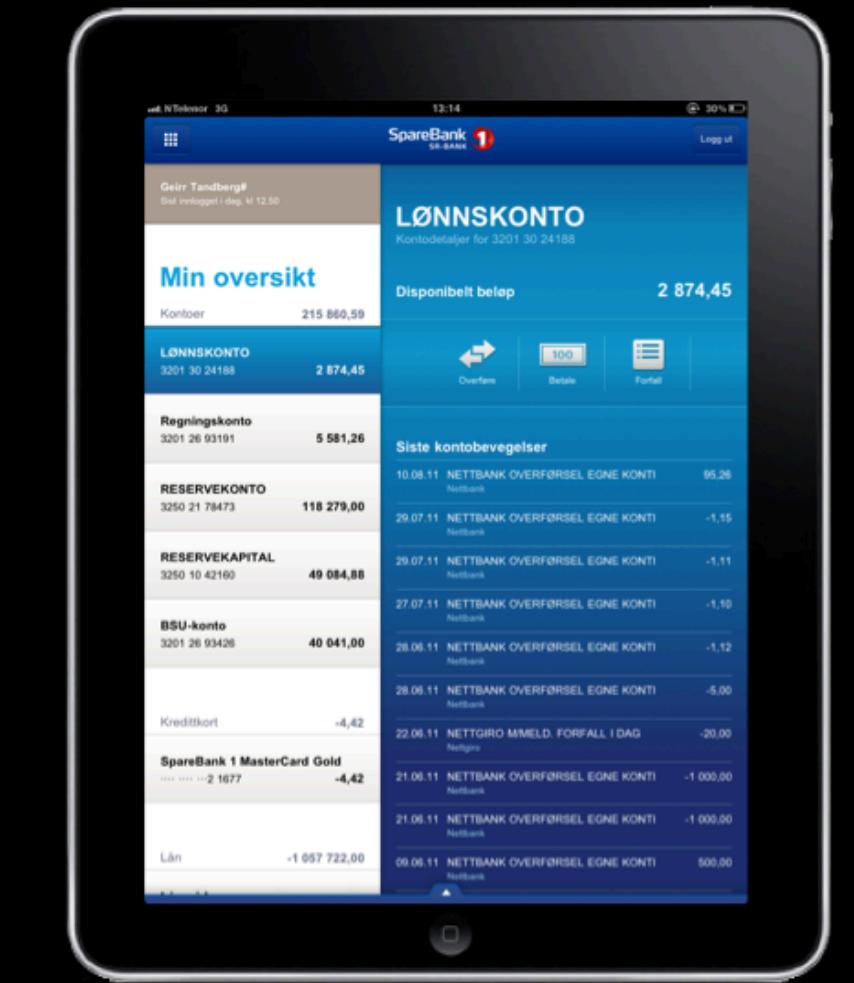
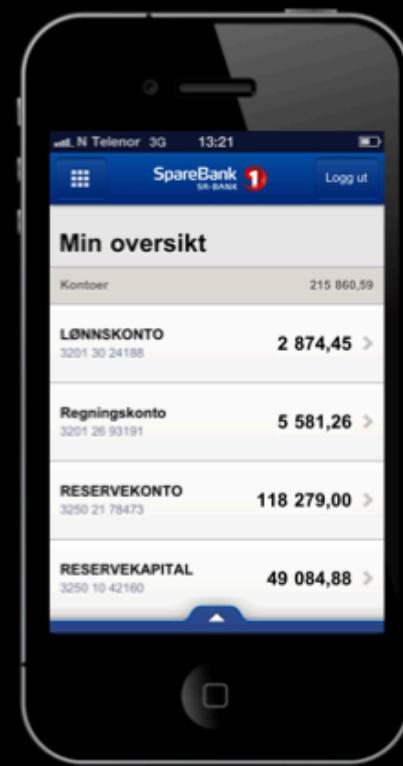
APPROX 2 YEARS OF EXPERIENCE WITH REACT

CURRENT PROJECT: MOBILE BANKING APP

THE BEAST IN NUMBERS



- > 8M MONTHLY LOGINS
- > ROUGHLY 500K USERS
- > OVER 100K LINES OF JS
(BACKBONE.JS)





HIDDEN COMPLEXITY

- > FOUR METHODS OF AUTHENTICATION
 - > TWO AUTHENTICATION LEVELS
 - > FEATURE SUBSET FOR MINORS (<18)
 - > FEATURE SUBSET FOR CHILDREN (<15)
 - > EVERY FEATURE IS TOGGLEABLE



WE NEED A PLAN. STAN

A photograph of a dense forest. Sunlight filters through the canopy of green leaves, creating bright highlights and deep shadows. The forest floor is visible at the bottom, showing more greenery and some fallen branches.

BACKBONE

IMPERATIVE FRAMEWORK FOR SPAs

REACT

DECLARATIVE LIB. DOM ONLY

THE COOL BIKE



THE COOL BIKE
BY AN ENGINEER

```
'account/:id': function(id) {  
  app.render(new AccountView({ id }));  
}
```

```
// ReactWrapperView.js
Backbone.View.extend({
  initialize({ view }) {
    React.render(view, this.el);
  },
  destroy() {
    React.unmountComponentAtNode(this.el);
  }
})
```

```
'account/:id': function(id) {  
  var wrapper = new ReactWrapperView({  
    view: <AccountView accountId={ id } />  
  });  
  app.render(wrapper);  
}  
}
```

```
'account/:id': function(id) {  
  app.render(wrap(<AccountView id={ id } />));  
}  
}
```

CODE EXAMPLES

HOW SMART SHOULD COMPONENTS BE?

ROUTE HANDLERS

```
'cards/credit/:id': function(cardId) {  
  app.render(<Spinner />);  
  
  creditCards.fetch()  
    .then(render)  
    .fail(onError)  
}
```

SMART COMPONENTS

```
export default React.createClass({
  componentDidMount() {
    this.props.cards.fetch().then(this.forceUpdate)
  },
  render() {
    // handle everything here
    ...
  }
})
```

SMART
VS
DUMB

SMART COMPONENTS

```
export default React.createClass({
  componentDidMount() { /* fetch data */ },
  getInitialState() { return { ... } },
  submit() {
    // validation
    // action
  },
  render() {
    return <PaymentForm {...this.state} submit={this.submit} />
  }
});
```

DUMB COMPONENTS

```
export default React.createClass({
  render() {
    return <div>
      <AccountSelector ... />
      <MessageField ... />
      ...
      <button onClick={ this.props.submit } />
    </div>;
  }
});
```

THE STATE MACHINE

HOW TO HANDLE ALL OUTCOMES

THE STATE MACHINE

HOW TO HANDLE ALL OUTCOMES

- > IS FETCHING
- > HAS RESPONSE
- > HAS FAILED

```
render() {  
  ...  
  if (awaitingResponse)  
    return <Spinner />;  
  
  if (debitCardsFailed)  
    return <DebitCardsFailureMessage />;  
  
  return <PaymentForm {...this.props} />  
}
```

HANDLING USER INPUT

COMPLEXITY COMES WITH MANY FORMS

SpareBank
SMN 1
nsa50247

< Avbryt

Betale

Fra konto

4202 59 43990 BRUKSKONTO FRI CARL J ▾

Disponibelt beløp 1,48

Til konto

Forfallsdato

dd.mm.åååå **I dag**

Beløp

,

KID-nummer **Melding**

Oppgi KID-nummer

Jeg ønsker kvitteringsoblat for betalingen

Legg til godkjenning Avbryt

MANUAL LABOR 2.0

```
events: {  
  '.message' : 'handleMessageClick'  
},  
handleMessageClick(e) {  
  this$('.cid').hide();  
  this$('.message').show();  
  // do some other stuff  
}  
  
<input class="message" ... />
```

MANUAL LABOR 2.0

```
events: {  
  '.message' : 'handleMessageClick',  
  '.toAccount' : 'toAccountChanged'  
},  
handleMessageClick(e) {  
  this$('.cid').hide();  
  this$('.message').show();  
  ...  
},  
toAccountChanged(e) {  
  ...  
}  
;
```

```
<input class="message" ... />  
<input class="toAccount" ... />
```

```
updateAmount(evt) {  
    // custom handling of this event  
}
```

```
<input onChange={ this.updateAmount } />
```

```
function updateState(key) {  
  return function(e) {  
    this.setState({ [key]: e.target.value })  
  }  
}  
  
<select onChange={ updateState('fromAccount') } ... />  
  
<input onChange={ updateState('amountInteger') } />
```

```
function updateState(key, value) {  
  return function(e) {  
    this.setState({ [key]: value || e.target.value })  
  }  
}  
  
<button onClick={ updateState('message', true) } ... />
```

```
function updateState(key, value) {
  return function(e) {
    this.setState(typeof key === 'object' ?
      key :
      { [key]: value || e.target.value })
  }
}

<button onClick={

  updateState({ 'message': true, 'cid': false })

} />
```

READABILITY
AND
REUSABILITY

REUSABILITY AND READABILITY

```
initialize() {  
    this.fromAccountSelector = new AccountSelector(options)  
},  
render() {  
    this.renderTemplate();  
    this.fromAccountSelector.setElement(this.$('.from-account'));  
    this.fromAccountSelector.render();  
}
```

REUSABILITY AND READABILITY

```
render() {  
  return <div>  
    ...  
    <AccountSelector ... />  
  </div>  
}
```

CONDITIONAL RENDERING

CONDITIONAL RENDERING

```
render() {  
  return <div>  
    { someLogic ? <div>Some text<div> : '' }  
  </div>;  
}
```

CONDITIONAL RENDERING

```
render() {  
  
  const oneTree = <div>  
    <h1>Item list</h1>  
    { this.props.items.map(toListElement) }  
    More stuff.  
  </div>;  
  
  const otherTree = ...  
  
  const shownTree = someCondition ? oneTree : otherTree;  
  
  return <div> { shownTree }</div>;  
}
```

CONDITIONAL RENDERING

```
render() {  
  return <div>  
    <If condition={ someBooleanExpression }>  
      ...  
    </If>  
  </div>;  
}
```



MUTATING STATE



VIEWS MUTATES MODEL DIRECTLY

IN A REDUX WORLD

```
submit() {  
  ...  
  if (validates) dispatch(action);  
}
```

LETS FAKE IT

```
submit() {  
    ...  
    if (validates) eventBus.trigger('action', data);  
}
```

LETS FAKE IT

```
eventBus.on('action', paymentAction);

function paymentAction(data) {
    // create new payment
    // save
    // invalidate & refetch singleton
}
```

SUMMARY



THANKS!

@EWNDL