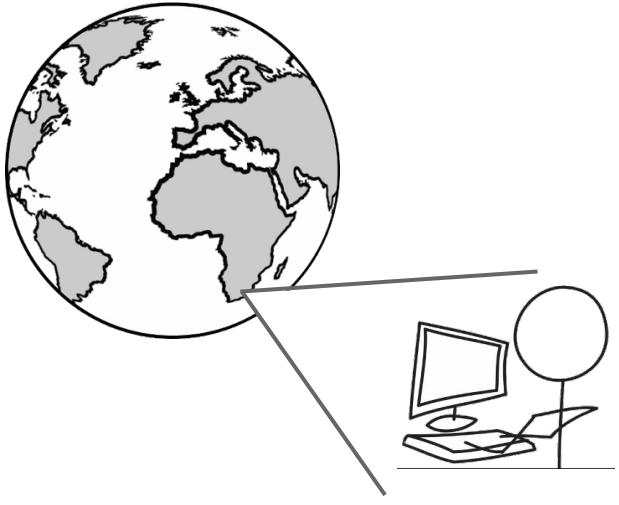
the Python-only web framework

Iwan Vosloo

About



10

z
o
p
e

80

reahl™

26 600 loc
(40% tests)

15 Python

Financial industry / in-house development

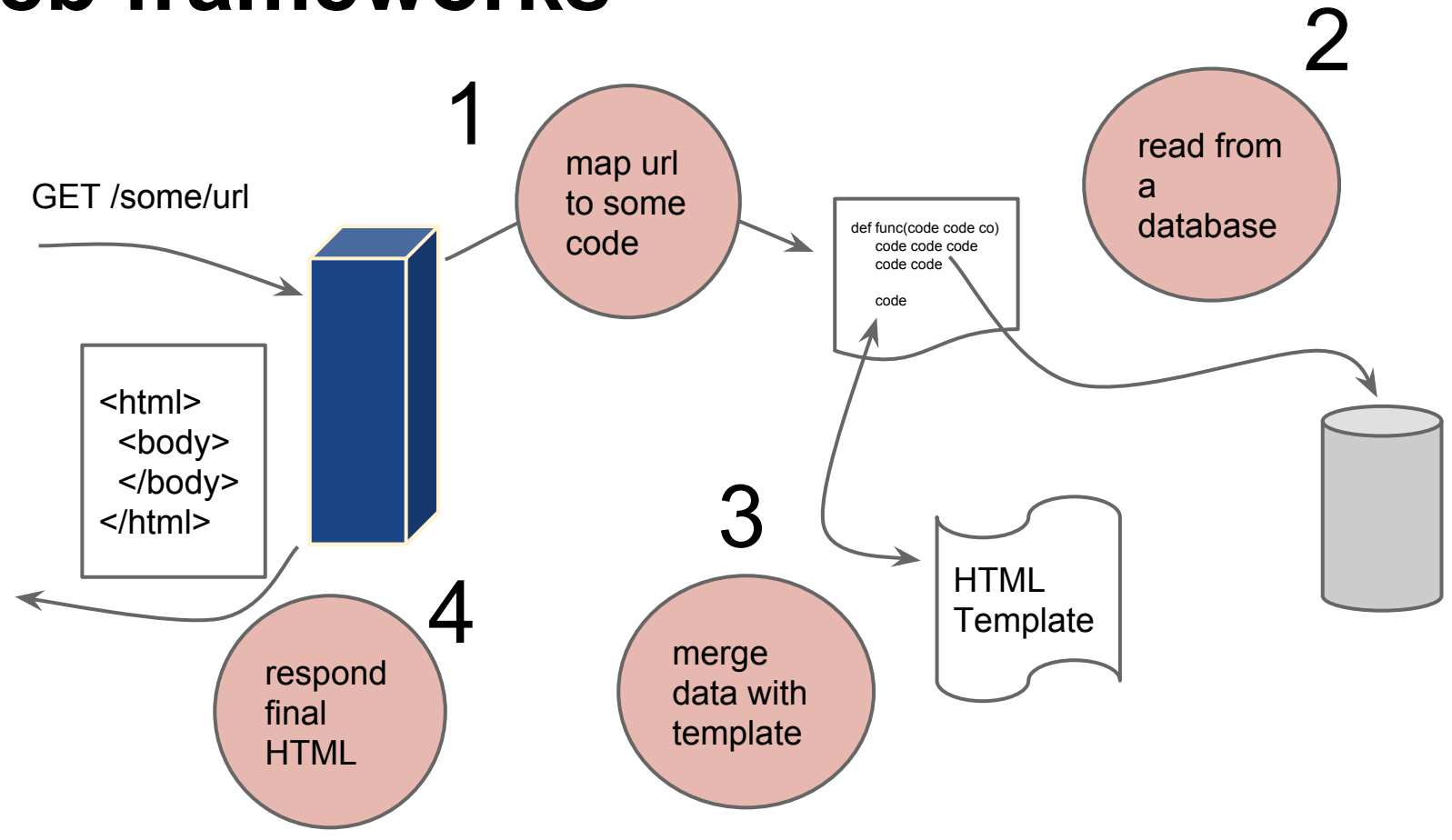
reahl™

the Python-only web framework

- Background
- What it is
- How it works
- Why you should care
- Strategy
- Status
- Questions

Background

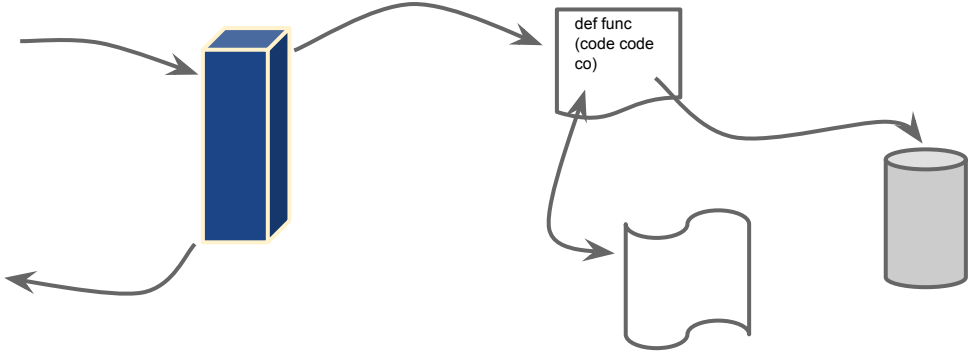
Web frameworks



Background

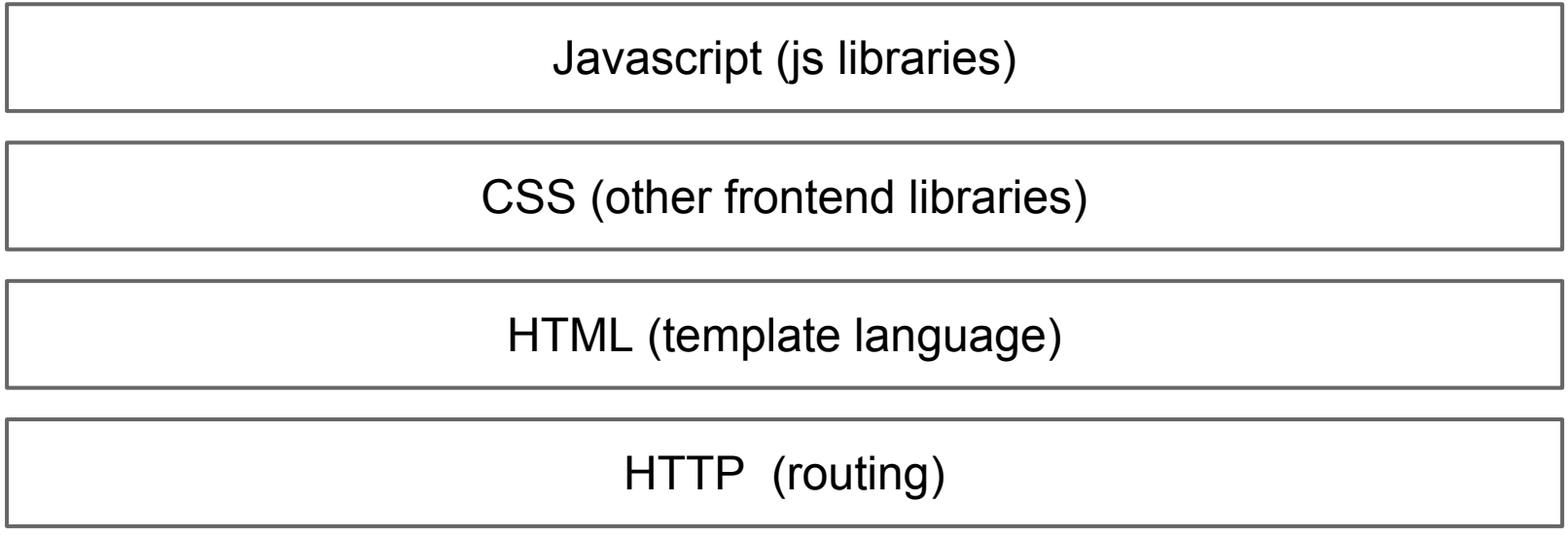
Web frameworks

re-use templates | css frameworks | js libraries



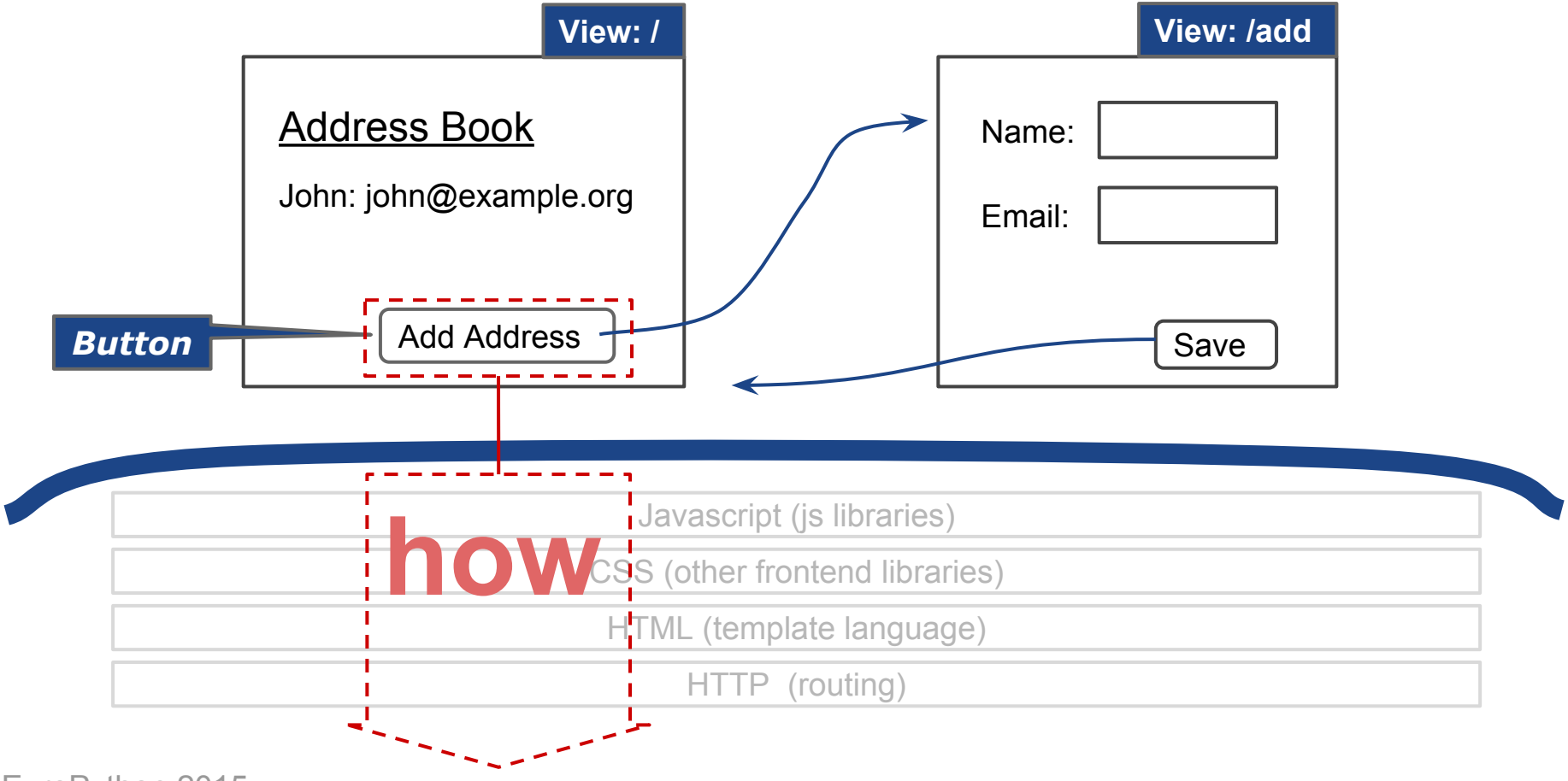
Background

Technology focus



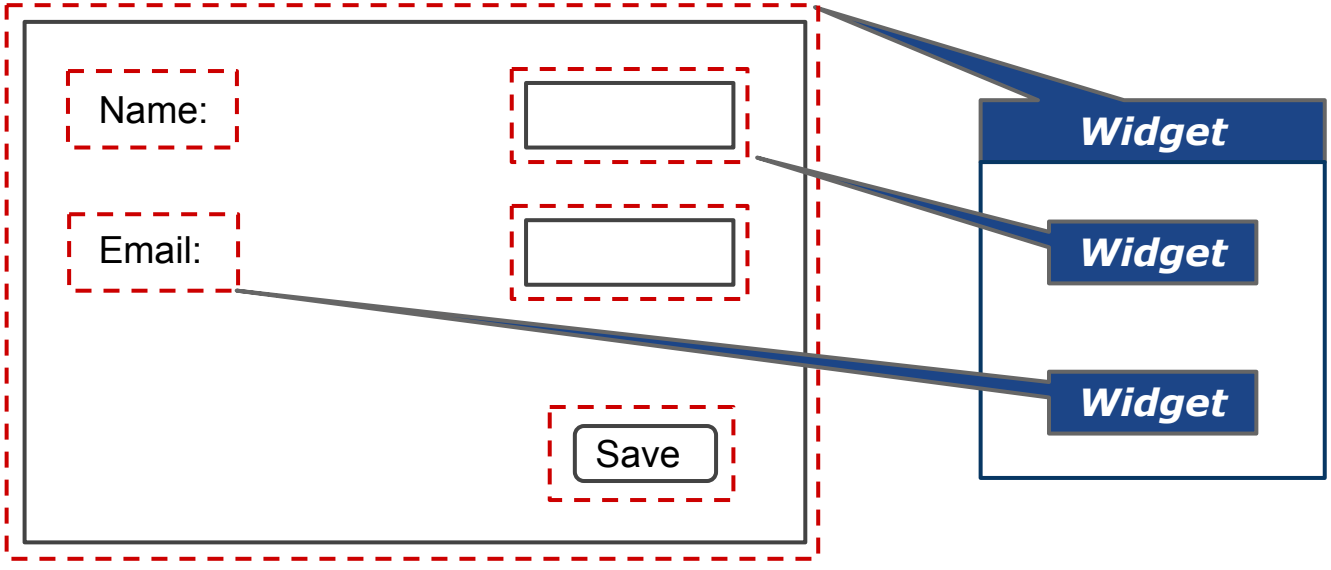
What it is

Goal focus



What it is

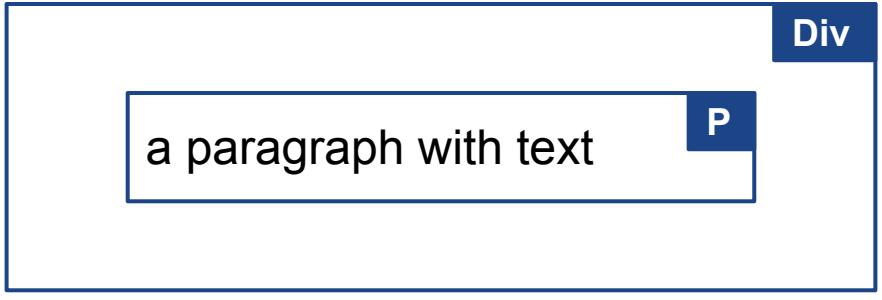
Pages are widgets



What it is

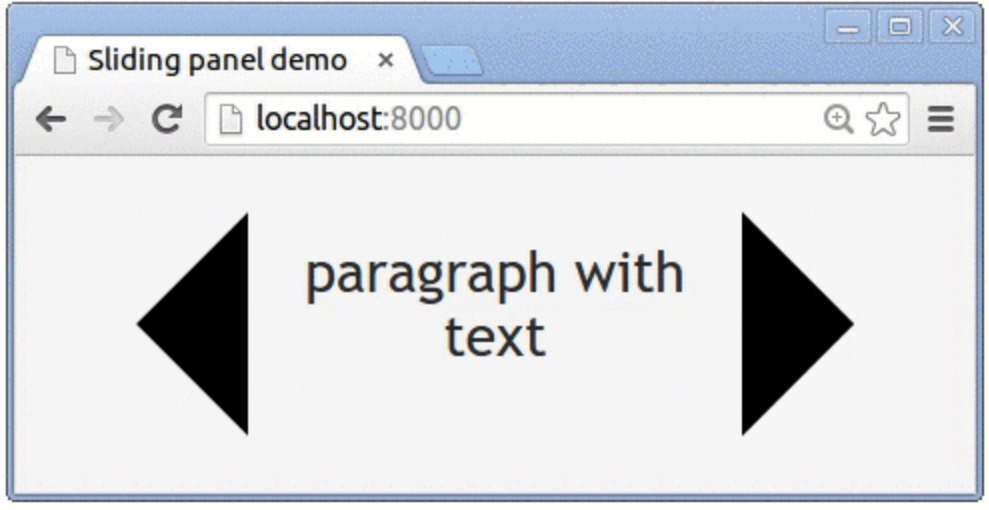
Composing widgets

```
panel1 = Div(view)
panel1.add_child( P(view, text='a paragraph with text') )
```



What it is

Widgets are not HTML



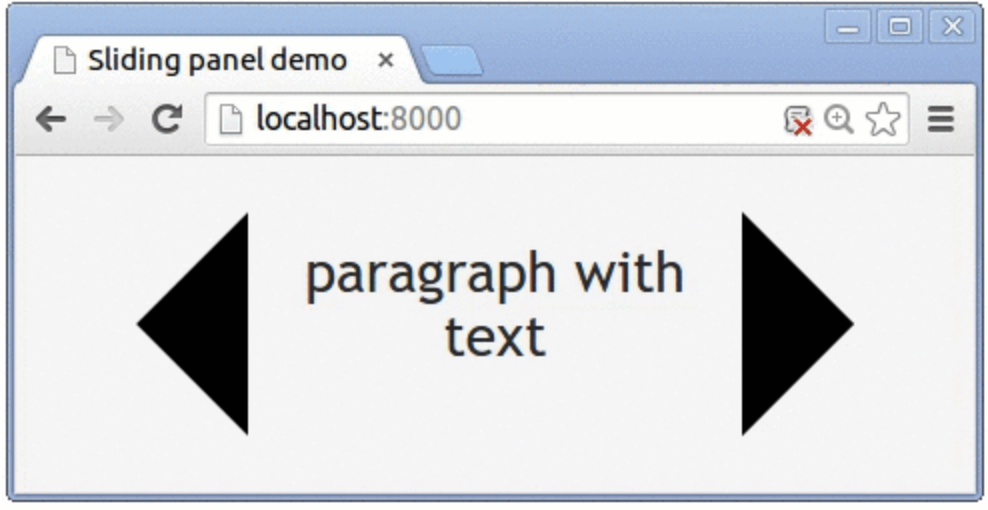
What it is

Widgets are not HTML

```
sliding_panel = SlidingPanel(view)

panel1 = sliding_panel.add_panel(Div(view))
panel1.add_child(P(view, text='a paragraph with text'))

panel2 = sliding_panel.add_panel(Div(view))
panel2.add_child(P(view, text='a different paragraph'))
```

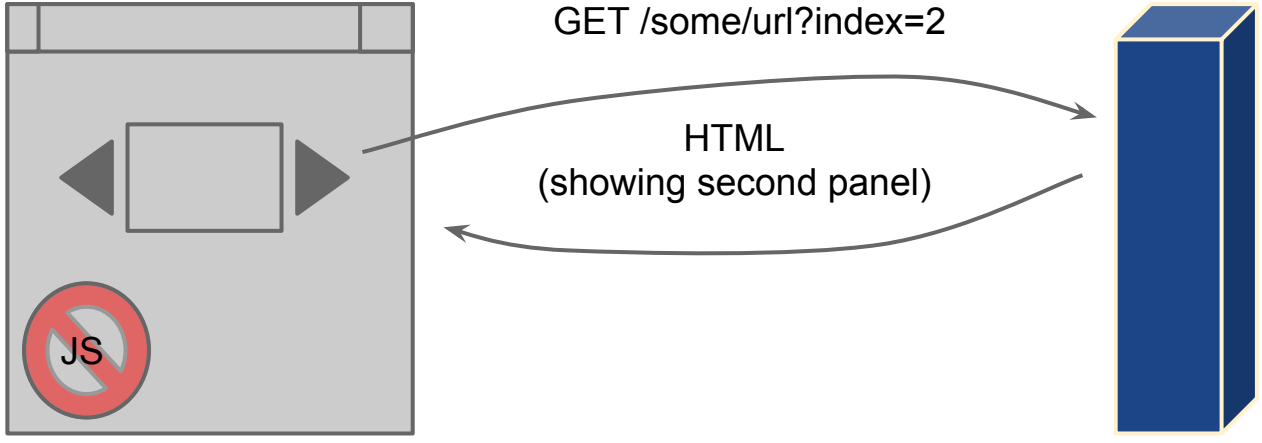
What it is

Widgets are not HTML



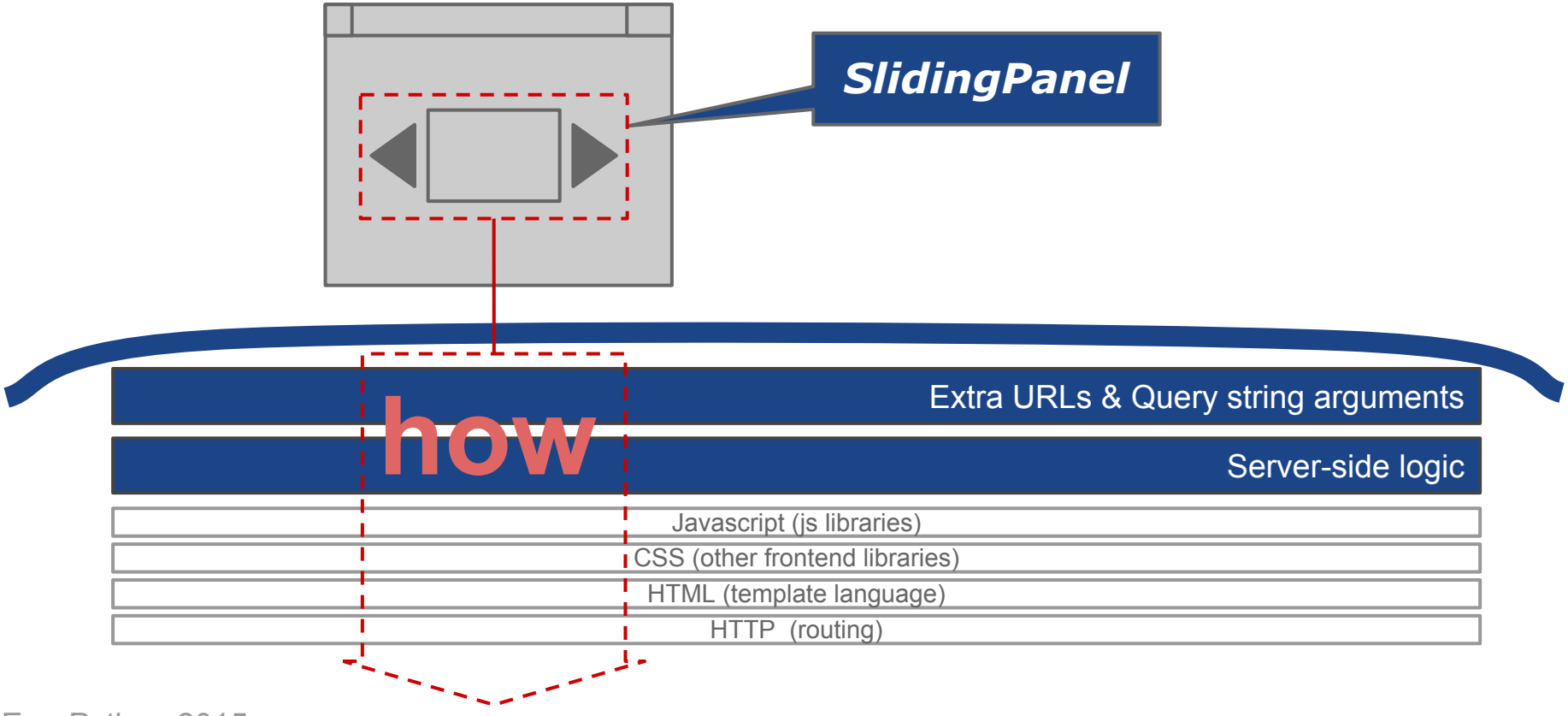
How it works

Behind the SlidingPanel



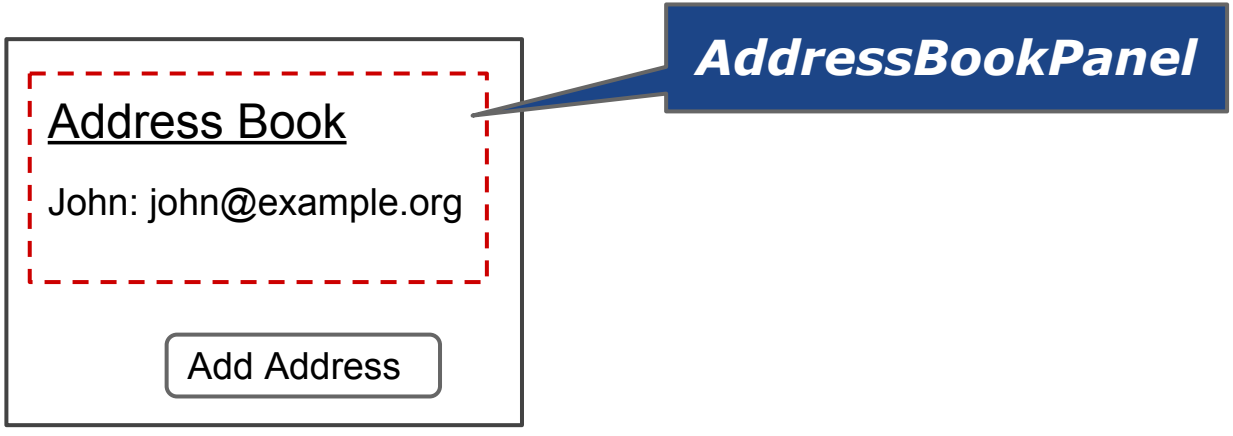
How it works

Behind the SlidingPanel



How it works

Widget lifetime



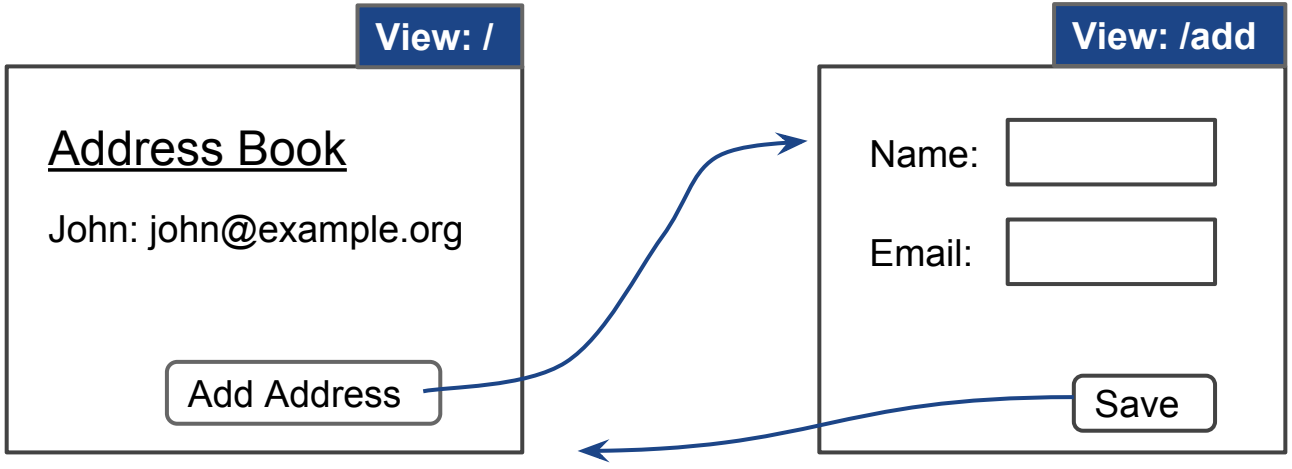
How it works

Widget lifetime

```
class AddressBookPanel(Div):
    def __init__(self, view):
        super(AddressBookPanel, self).__init__(view)

        self.add_child(H(view, 1, text='Address Book'))

        for address in Session.query(Address).all():
            self.add_child(AddressBox(view, address))
```

How it works

Defining a UserInterface



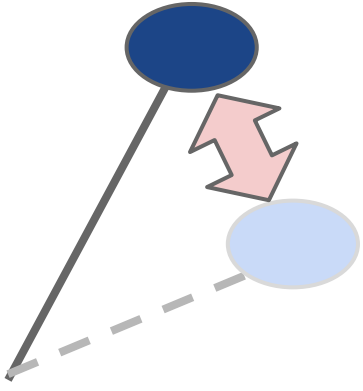
How it works

Defining a UserInterface

```
class AddressBookUI(UserInterface):

    def assemble(self):

        addresses = self.define_view('/', title='Addresses')
        addresses.set_page(HomePage.factory())

        add = self.define_view('/add', title='Add an address')
        add.set_page(AddPage.factory())

        self.define_transition(Address.events.save, add, addresses)
```

reahl™

Why you should care



Why you should care

Knowledge & focus

reahl™

Widget

Event

Button

Action

TextInput

Field

Layout

CheckboxInput

TabbedPanel

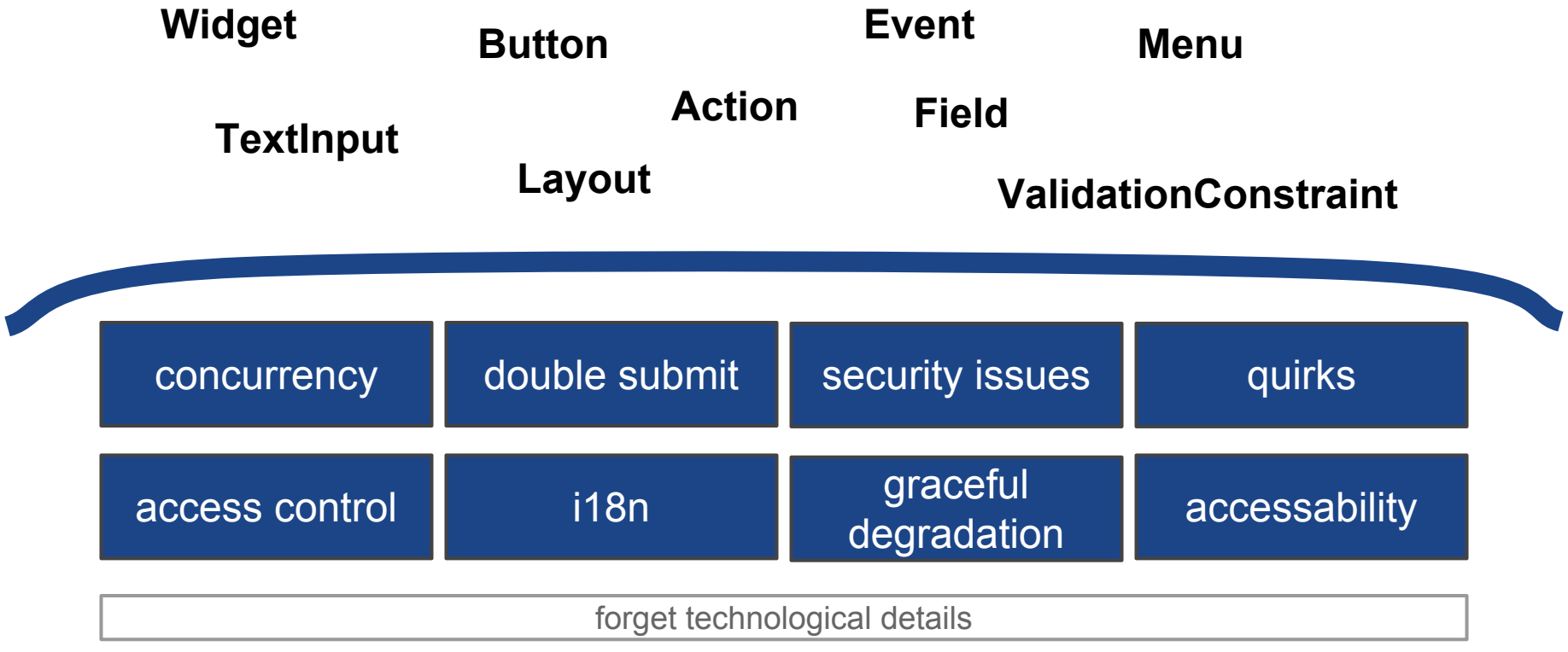
ValidationConstraint

Menu

forget technological details

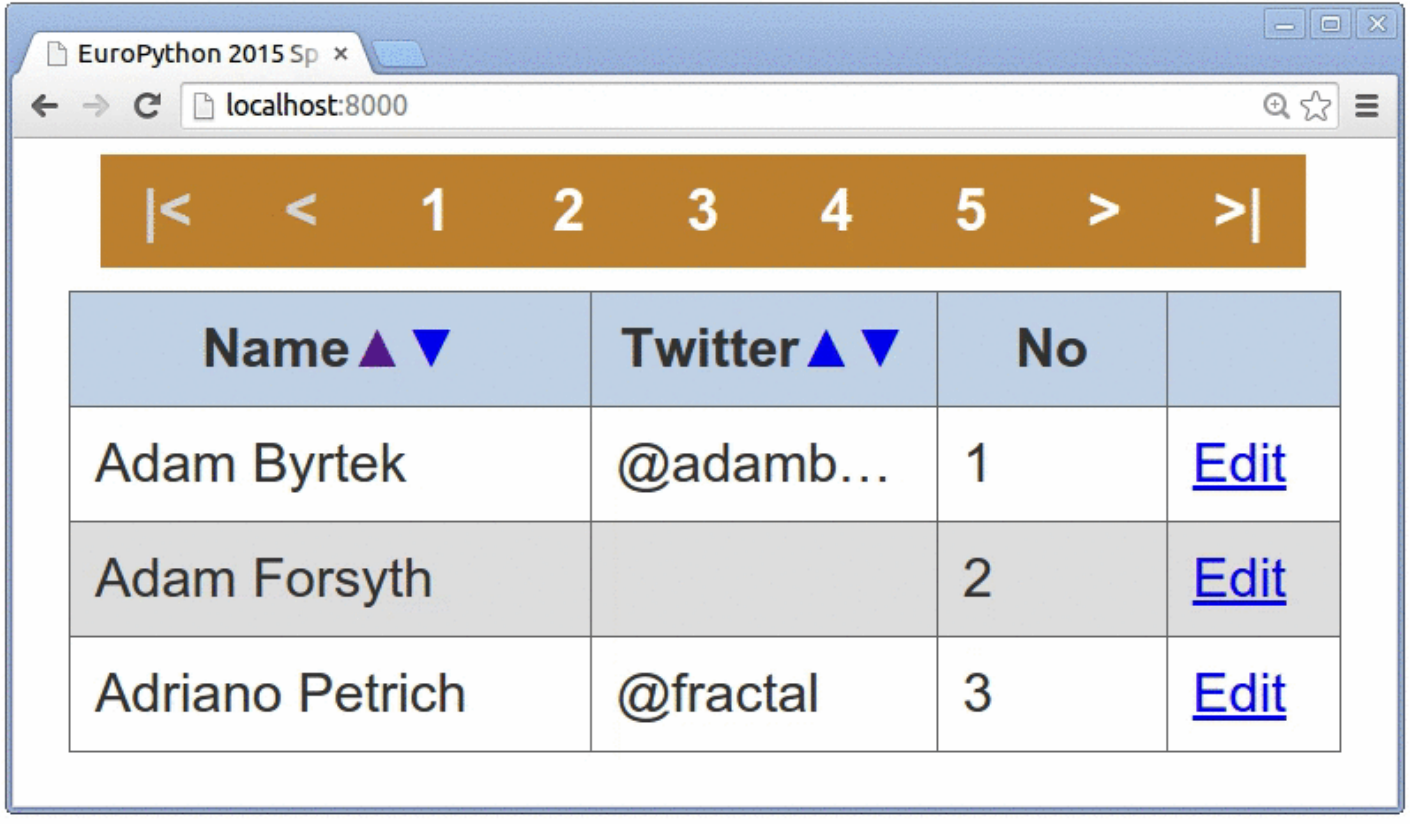
Why you should care

Re-use



Why you should care

Complicated requirements

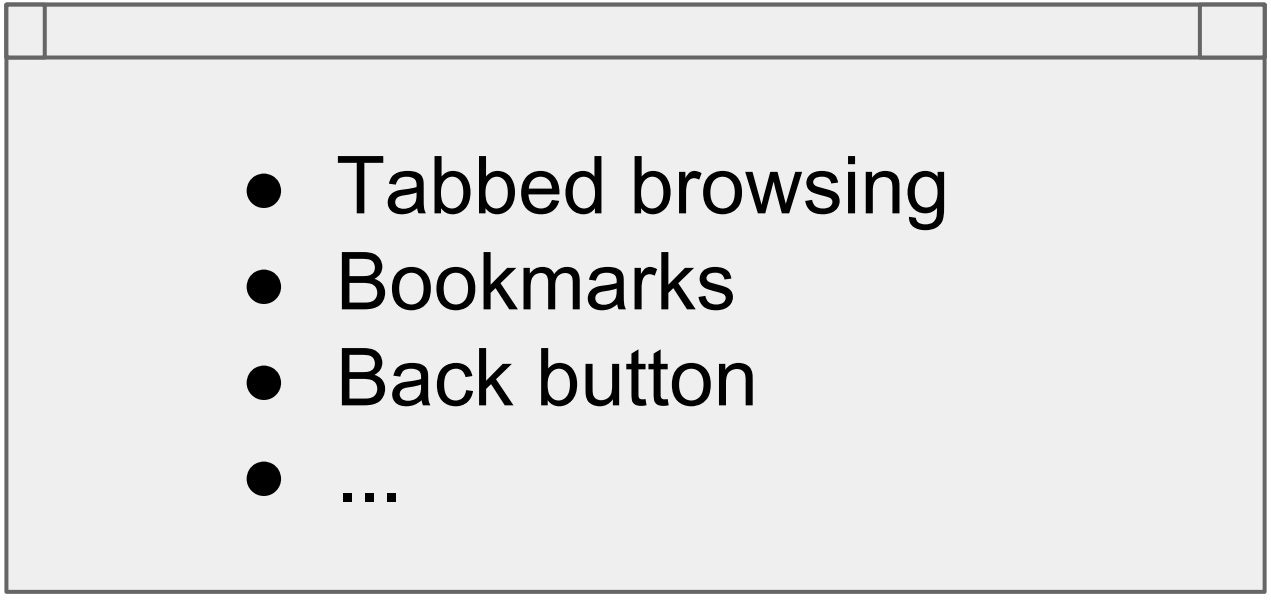


Strategy

“No-html” frameworks

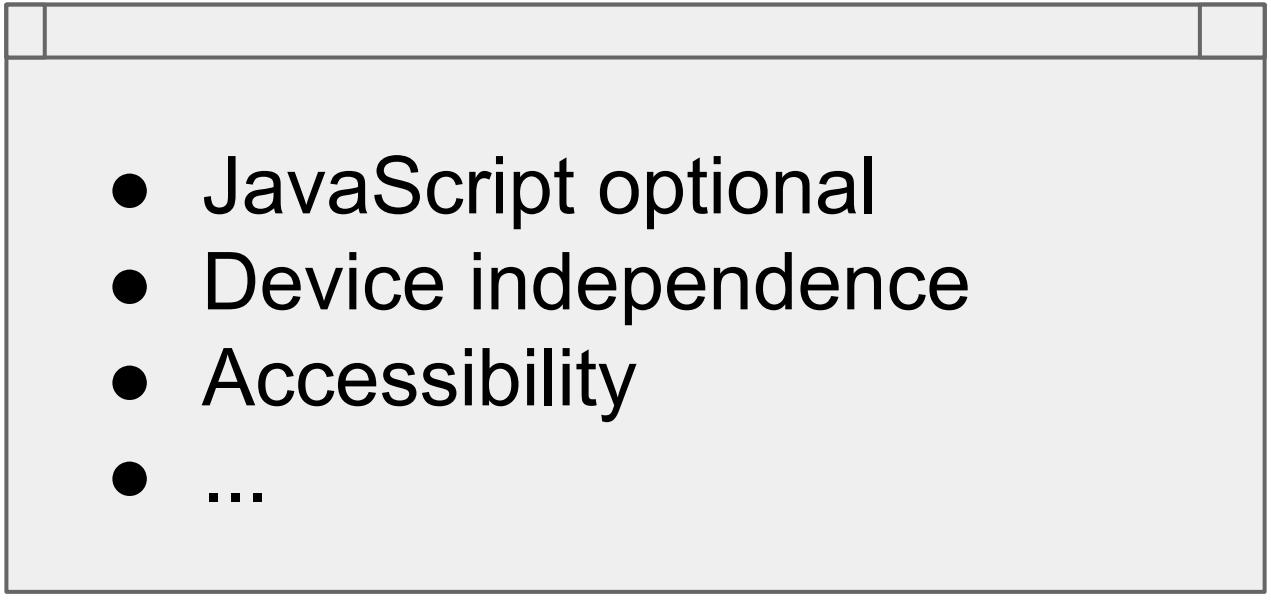
Strategy

Maintain web semantics



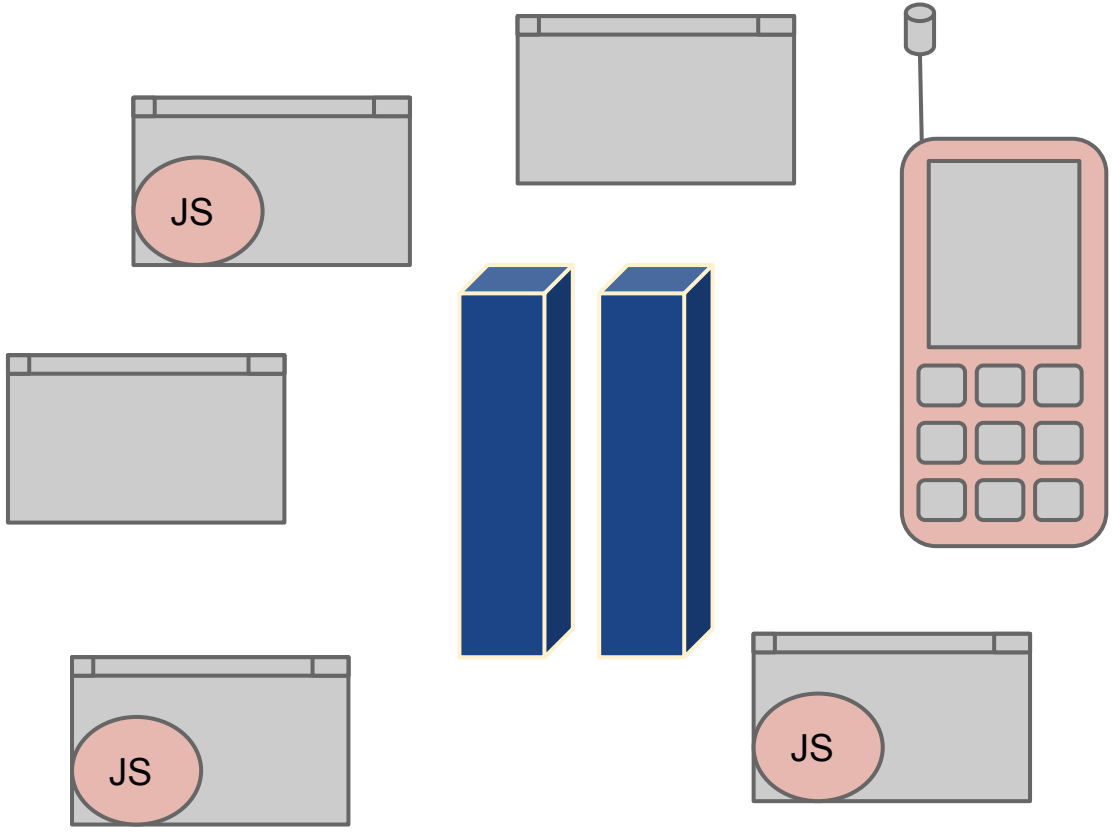
Strategy

Support web ideals



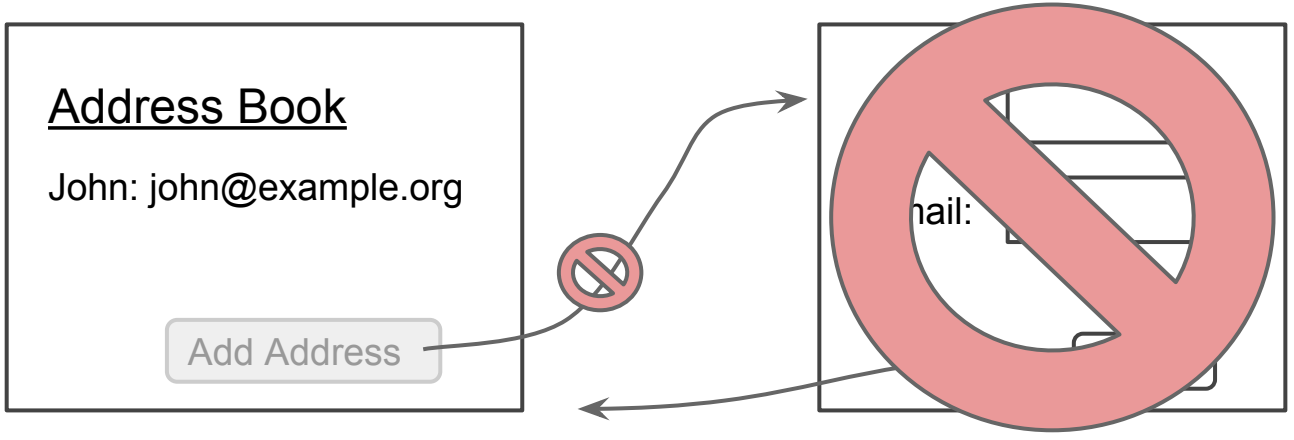
Strategy

Web-grown API



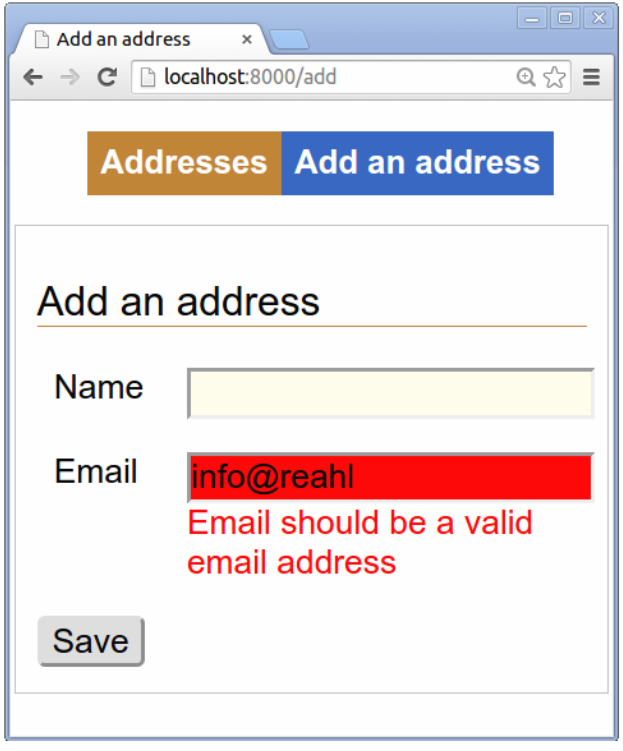
Strategy

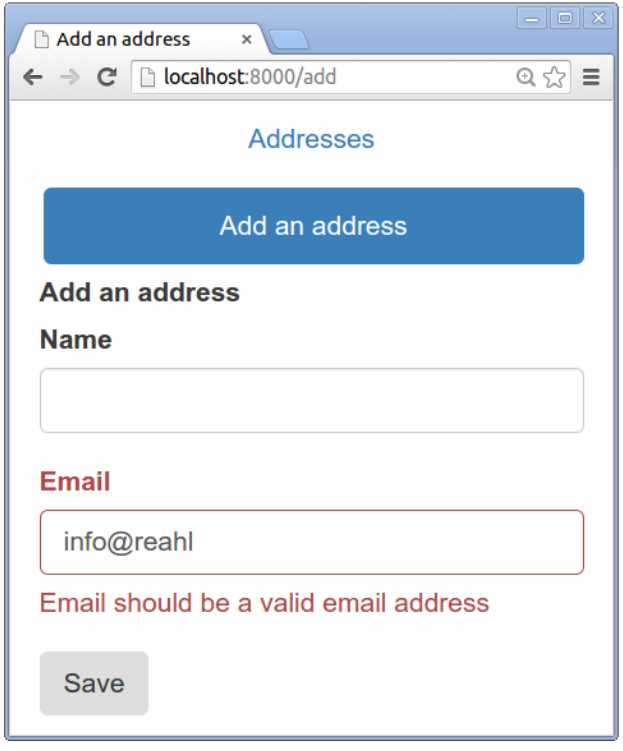
Aim: higher level issues



Strategy

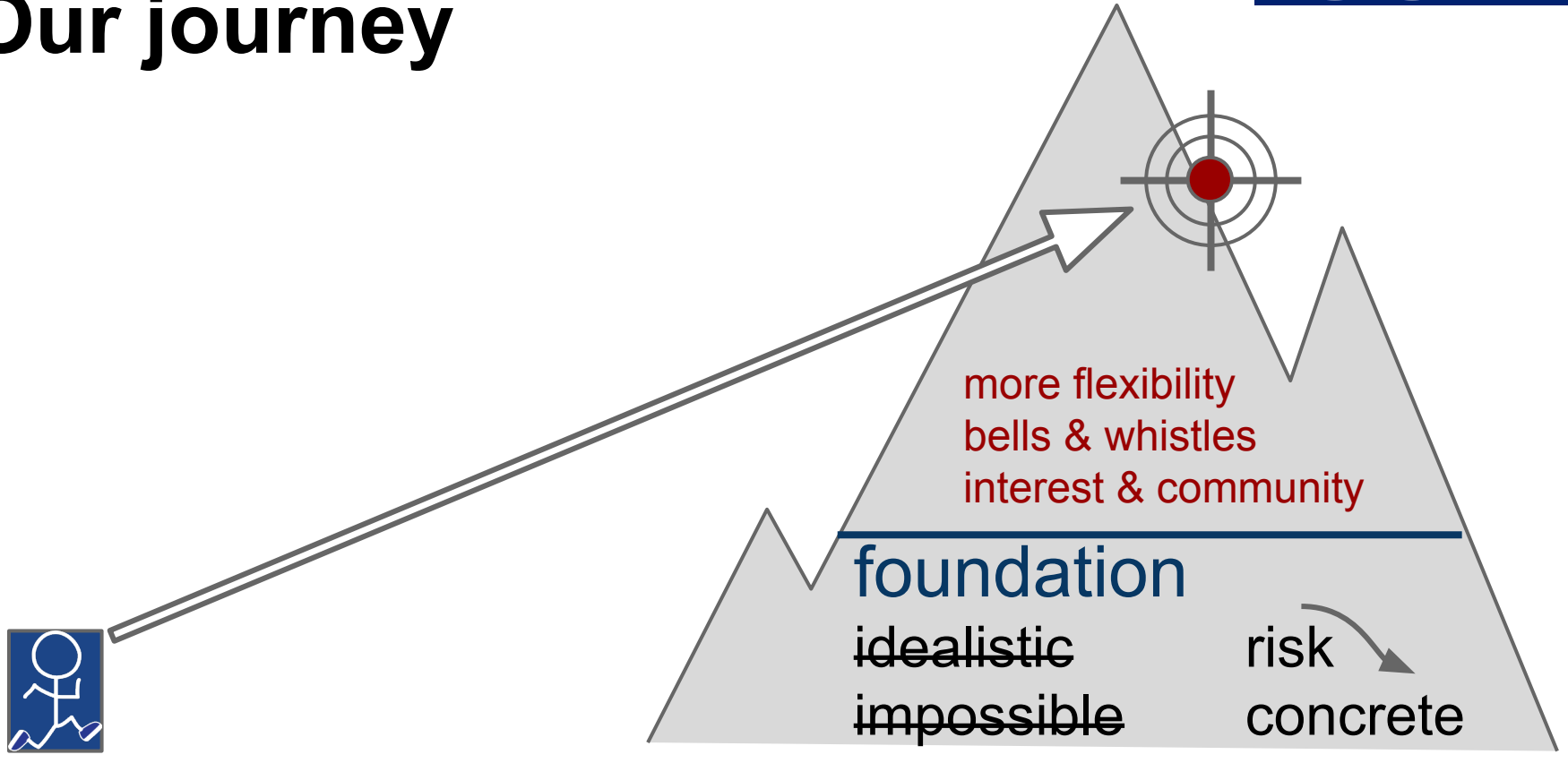
Leverage existing wisdom





Status

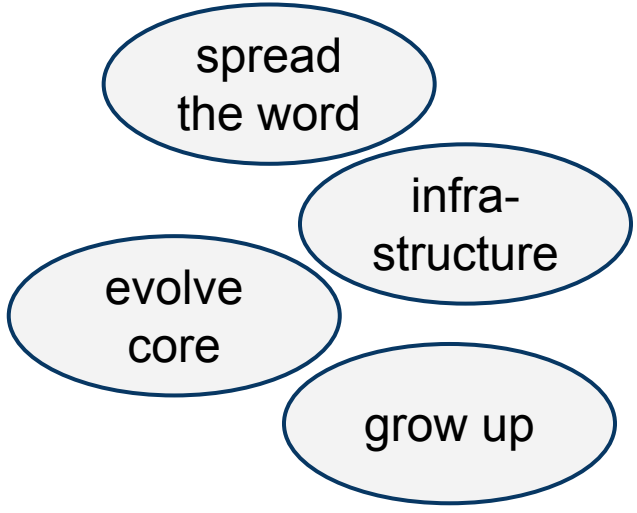
Our journey



Dream

You're invited

1. Look at examples
2. Join mailing list(s)
3. Install
4. Follow tutorial
5. Help



Thanks

www.reahl.org

@reahl

On **groups.google.com**:

- reahl-announce
- reahl-discuss

iwan@reahl.org