# VivideJS

Jonas Chromik
Web-based Development Environments
Winter Term 2017/2018

Software Architecture Group
Hasso Plattner Institute

# Outline

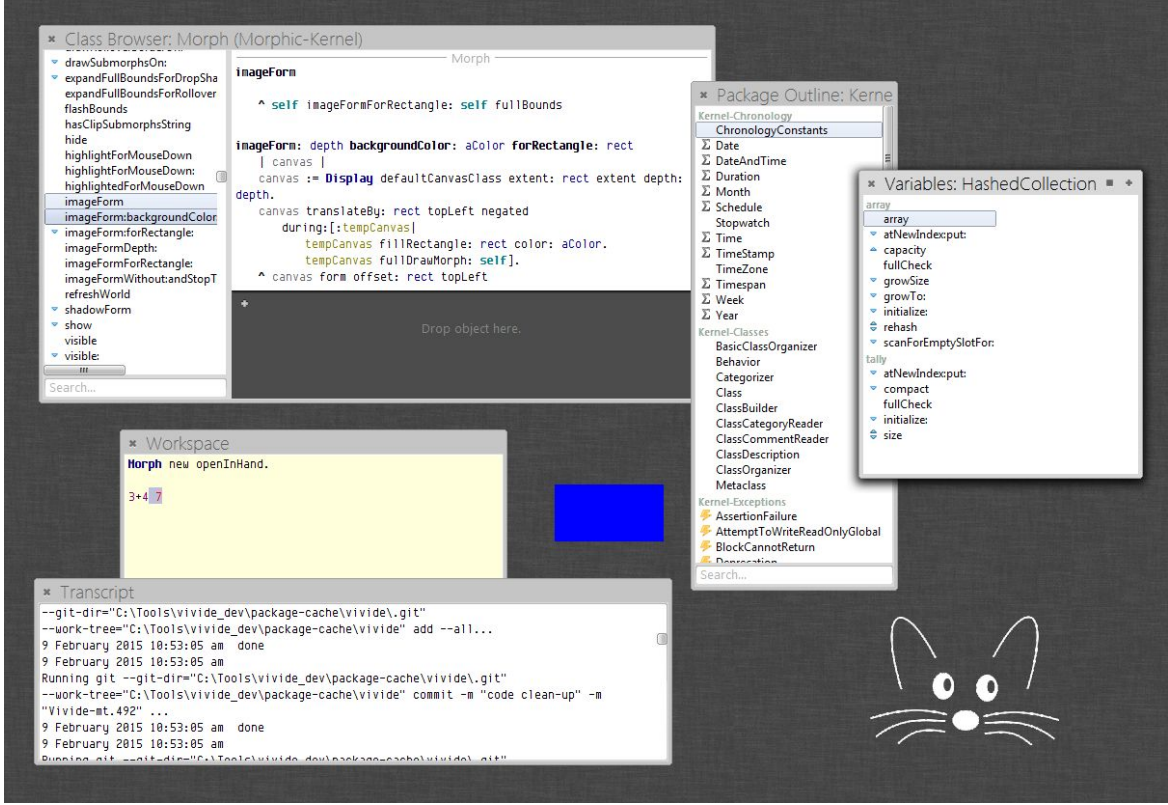
1. The Original Vivide
2. Structure of Widgets
3. Interaction between Widgets
4. Tooling
5. Demo

# The Original Vivide

# Vivide

Data-driven tool building environment by Marcel

Complex, feature rich, polished. Hence we may only cover a small share of it.



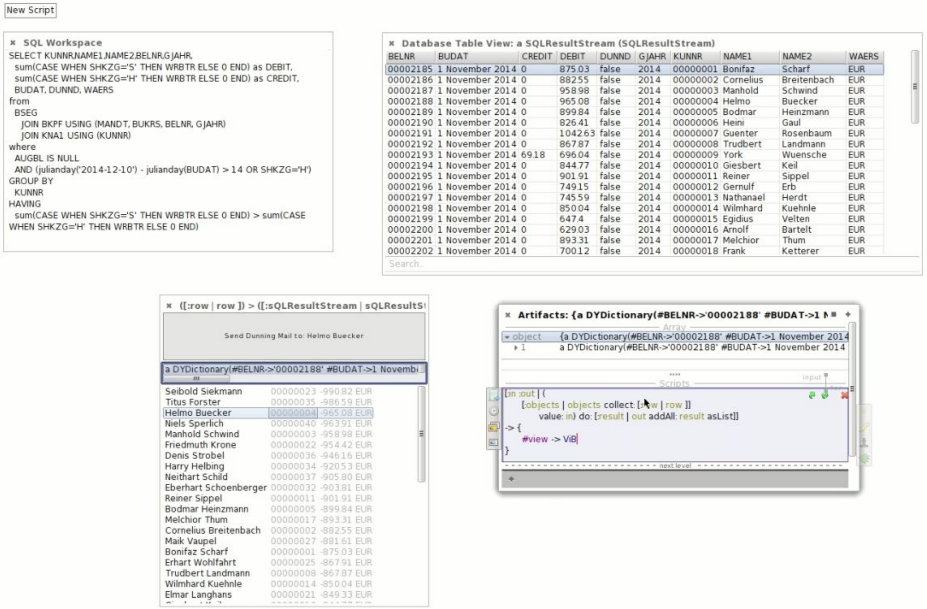

https://github.com/hpi-swa/vivide

# Vivide: Use Cases

In general: All kinds of data-displaying tools

Class Browser
File Browser
…

Guideline for VivideJS:
Patrick's Dunning Videos



Data Source

New Script

× SQL Workspace

```
SELECT KUNNR,NAME1,NAME2,BELNR,GJAHR,
  sum(CASE WHEN SHKZG='S' THEN WRBTR ELSE 0 END) as DEBIT,
  sum(CASE WHEN SHKZG='H' THEN WRBTR ELSE 0 END) as CREDIT,
  BUDAT, DUNND, WAERS
from
  BSEG
    JOIN BKPF USING (MANDT, BUKRS, BELNR, GJAHR)
    JOIN KNA1 USING (KUNNR)
where
  AUGBL IS NULL
  AND (julianday('2014-12-10') - julianday(BUDAT) > 14 OR SHKZG='H')
GROUP BY
  KUNNR
HAVING
  sum(CASE WHEN SHKZG='S' THEN WRBTR ELSE 0 END) > sum(CASE
WHEN SHKZG='H' THEN WRBTR ELSE 0 END)
```

× Database Table View: a SQLResultStream (SQLResultStream)

| BELNR | BUDAT | CREDIT | DEBIT | DUNND | GJAHR | KUNNR | NAME1 | NAME2 | WAERS |
|---|---|---|---|---|---|---|---|---|---|
| 00002185 | 1 November 2014 | 0 | 875.03 | false | 2014 | 00000001 | Bonifaz | Scharf | EUR |
| 00002186 | 1 November 2014 | 0 | 882.55 | false | 2014 | 00000002 | Cornelius | Breitenbach | EUR |
| 00002187 | 1 November 2014 | 0 | 958.98 | false | 2014 | 00000003 | Manhold | Schwind | EUR |
| 00002188 | 1 November 2014 | 0 | 965.08 | false | 2014 | 00000004 | Helmo | Buecker | EUR |
| 00002189 | 1 November 2014 | 0 | 899.84 | false | 2014 | 00000005 | Bodmar | Heinzmann | EUR |
| 00002190 | 1 November 2014 | 0 | 826.41 | false | 2014 | 00000006 | Heini | Gaul | EUR |
| 00002191 | 1 November 2014 | 0 | 1042.63 | false | 2014 | 00000007 | Guenter | Rosenbaum | EUR |
| 00002192 | 1 November 2014 | 0 | 867.87 | false | 2014 | 00000008 | Trudbert | Landmann | EUR |
| 00002193 | 1 November 2014 | 69.18 | 696.04 | false | 2014 | 00000009 | York | Wuensche | EUR |
| 00002194 | 1 November 2014 | 0 | 844.77 | false | 2014 | 00000010 | Giesbert | Keil | EUR |
| 00002195 | 1 November 2014 | 0 | 901.91 | false | 2014 | 00000011 | Reiner | Sippel | EUR |
| 00002196 | 1 November 2014 | 0 | 749.15 | false | 2014 | 00000012 | Gernulf | Erb | EUR |
| 00002197 | 1 November 2014 | 0 | 745.59 | false | 2014 | 00000013 | Nathanael | Herdt | EUR |
| 00002198 | 1 November 2014 | 0 | 850.04 | false | 2014 | 00000014 | Wilmhard | Kuehnle | EUR |
| 00002199 | 1 November 2014 | 0 | 647.4 | false | 2014 | 00000015 | Egidius | Velten | EUR |
| 00002200 | 1 November 2014 | 0 | 629.03 | false | 2014 | 00000016 | Arnolf | Bartelt | EUR |
| 00002201 | 1 November 2014 | 0 | 893.31 | false | 2014 | 00000017 | Melchior | Thum | EUR |
| 00002202 | 1 November 2014 | 0 | 700.12 | false | 2014 | 00000018 | Frank | Ketterer | EUR |

Search...

Table View

× ([:row | row ]) > ([:sQLResultStream | sQLResultS

Send Dunning Mail to: Helmo Buecker

a DYDictionary(#BELNR->'00002188' #BUDAT->1 Novembe

| | | | |
|---|---|---|---|
| Seibold Siekmann | 00000023 | -990.82 | EUR |
| Titus Forster | 00000035 | -986.59 | EUR |
| Helmo Buecker | 00000004 | -965.08 | EUR |
| Niels Sperlich | 00000040 | -963.91 | EUR |
| Manhold Schwind | 00000003 | -958.98 | EUR |
| Friedmuth Krone | 00000022 | -954.42 | EUR |
| Denis Strobel | 00000036 | -946.16 | EUR |
| Harry Helbing | 00000034 | -920.53 | EUR |
| Neithart Schild | 00000037 | -905.80 | EUR |
| Eberhart Schoenberger | 00000032 | -903.81 | EUR |
| Reiner Sippel | 00000011 | -901.91 | EUR |
| Bodmar Heinzmann | 00000005 | -899.84 | EUR |
| Melchior Thum | 00000017 | -893.31 | EUR |
| Cornelius Breitenbach | 00000002 | -882.55 | EUR |
| Maik Vaupel | 00000027 | -881.61 | EUR |
| Bonifaz Scharf | 00000001 | -875.03 | EUR |
| Erhart Wohlfahrt | 00000025 | -867.91 | EUR |
| Trudbert Landmann | 00000008 | -867.87 | EUR |
| Wilmhard Kuehnle | 00000014 | -850.04 | EUR |
| Elmar Langhans | 00000021 | -849.33 | EUR |

List View

× Artifacts: {a DYDictionary(#BELNR->'00002188' #BUDAT->1 N

Array

object {a DYDictionary(#BELNR->'00002188' #BUDAT->1 November 2014

1 a DYDictionary(#BELNR->'00002188' #BUDAT->1 November 2014

Scripts

input

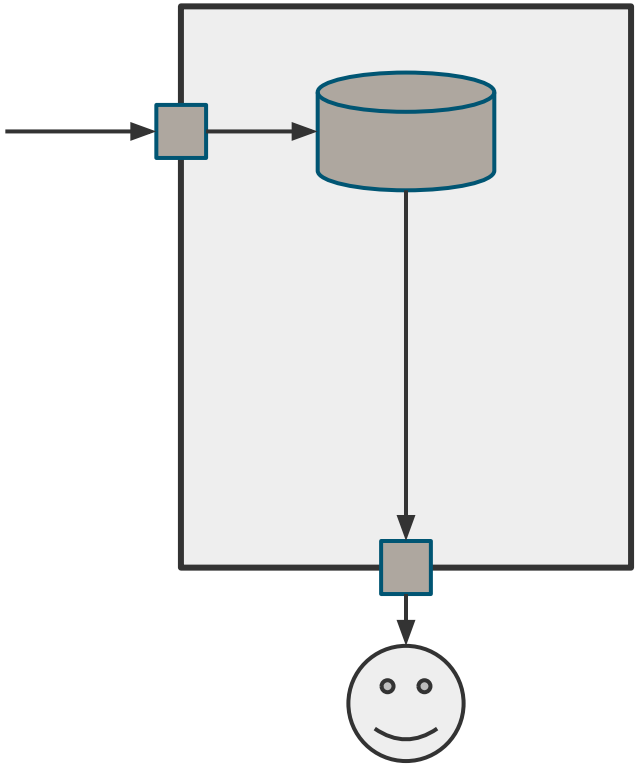
```
[in :out | (
    [objects | objects collect: [:row | row ]]
        value: in) do: [:result | out addAll: result asList]]
-> {
    #view -> ViB
}
```

next level

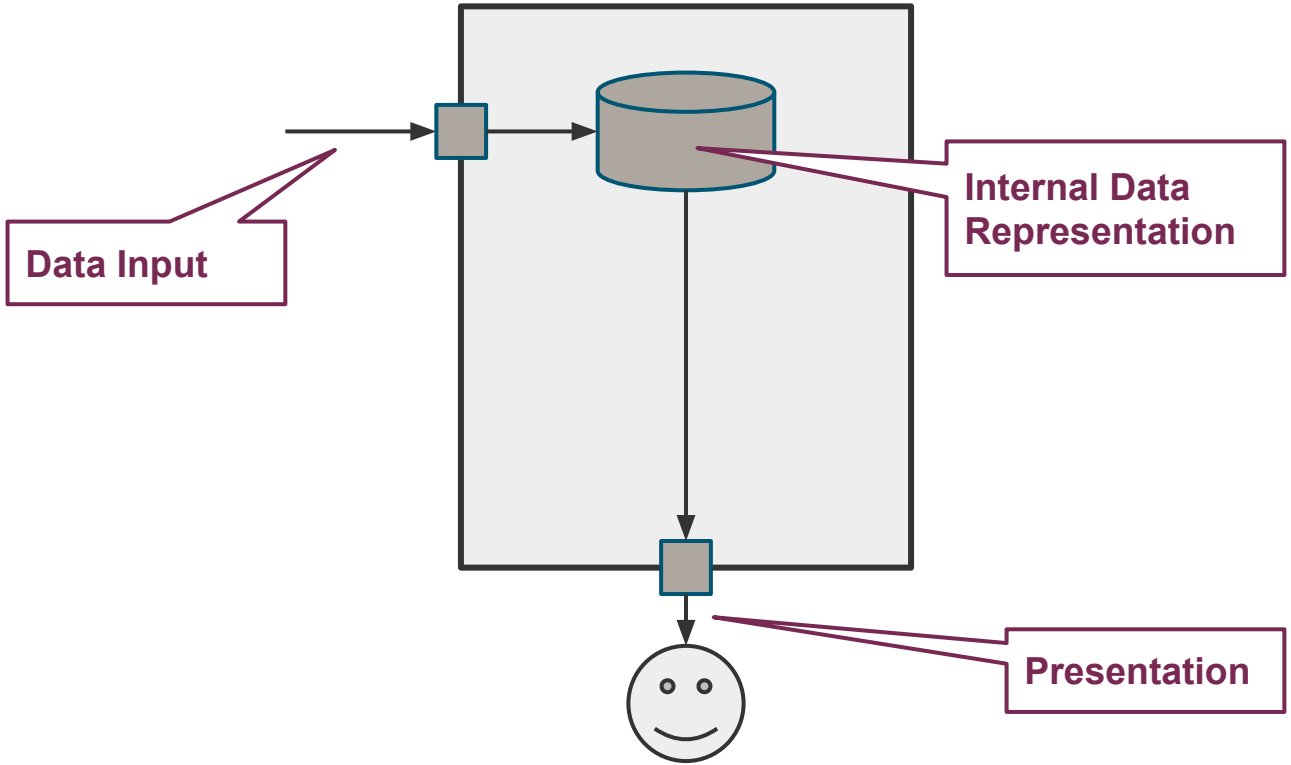
Inspector-ish tool called “Artifacts”

# Structure of Widgets

# Widgets: Basic Idea

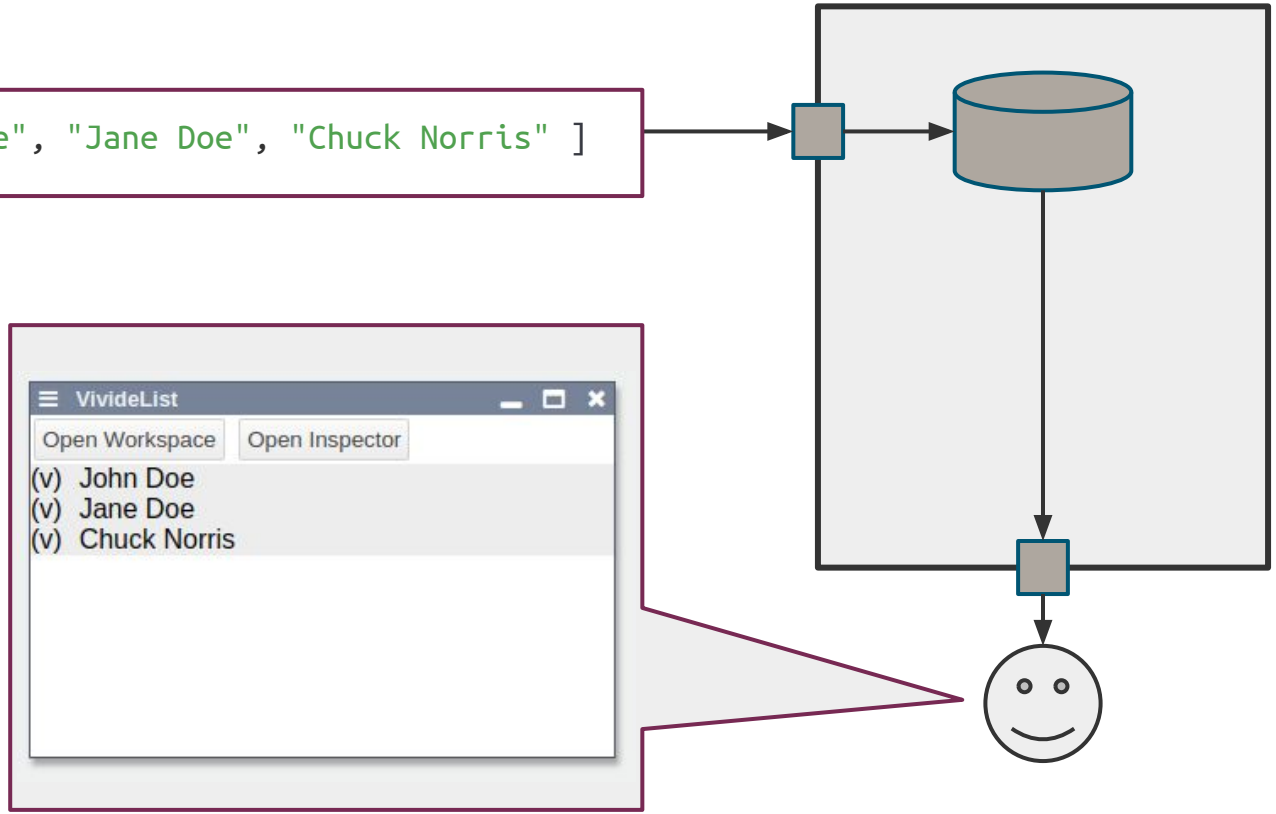


# Widgets: Basic Idea



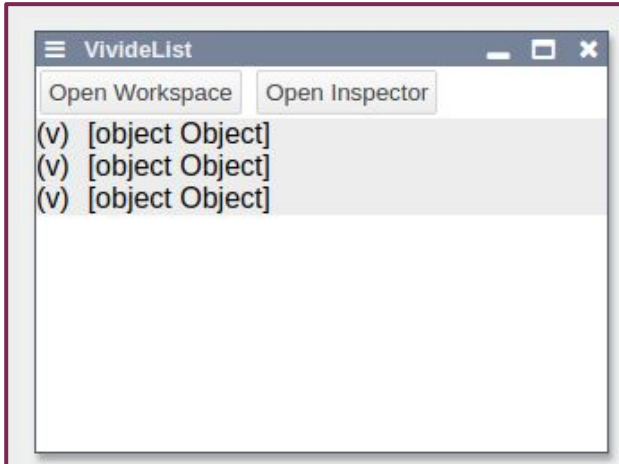
# Widgets: Basic Idea

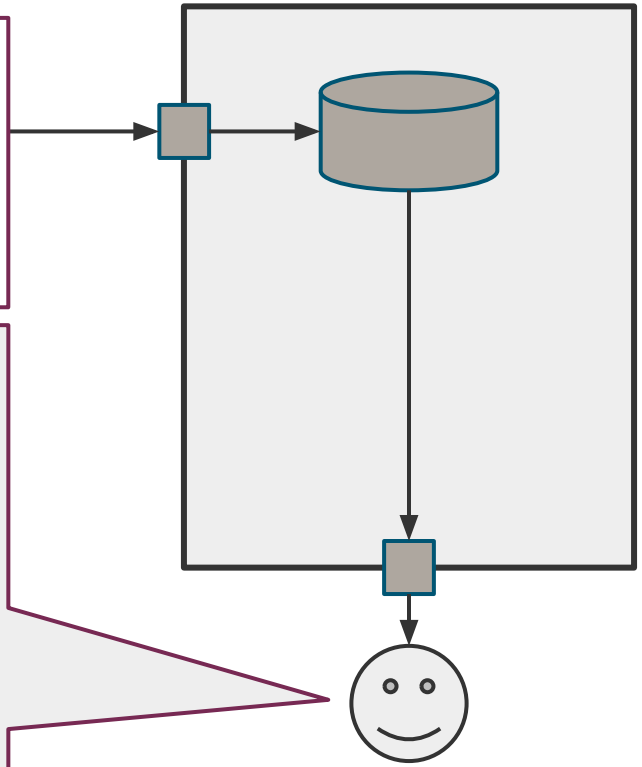
```
[ "John Doe", "Jane Doe", "Chuck Norris" ]
```



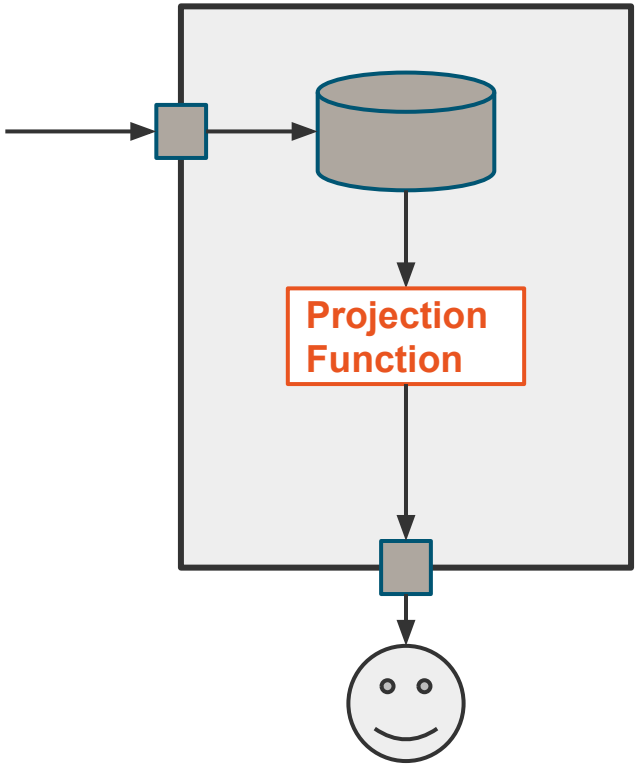
# Widgets: Basic Idea

```
[
 {name: "John Doe", age: 25, sex: "male", address: "New York"},
 {name: "Jane Doe", age: 24, sex: "female", address: "New York"},
 {name: "Chuck Norris", age: 77, sex: "male", address: "Everywhere"}
]
```



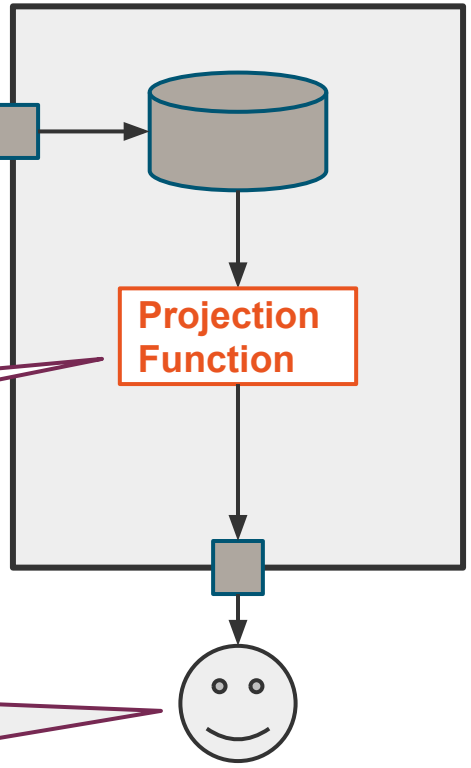



# Separating Data and Presentation



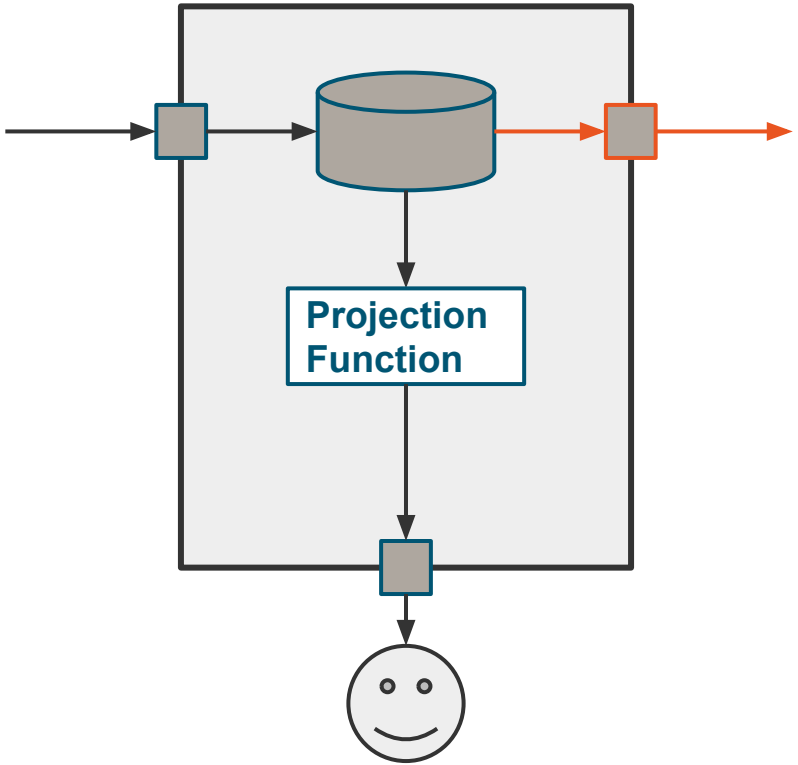
# Separating Data and Presentation

```
[
 {name: "John Doe", age: 25, sex: "male", address: "New York"},
 {name: "Jane Doe", age: 24, sex: "female", address: "New York"},
 {name: "Chuck Norris", age: 77, sex: "male", address: "Everywhere"}
]
```

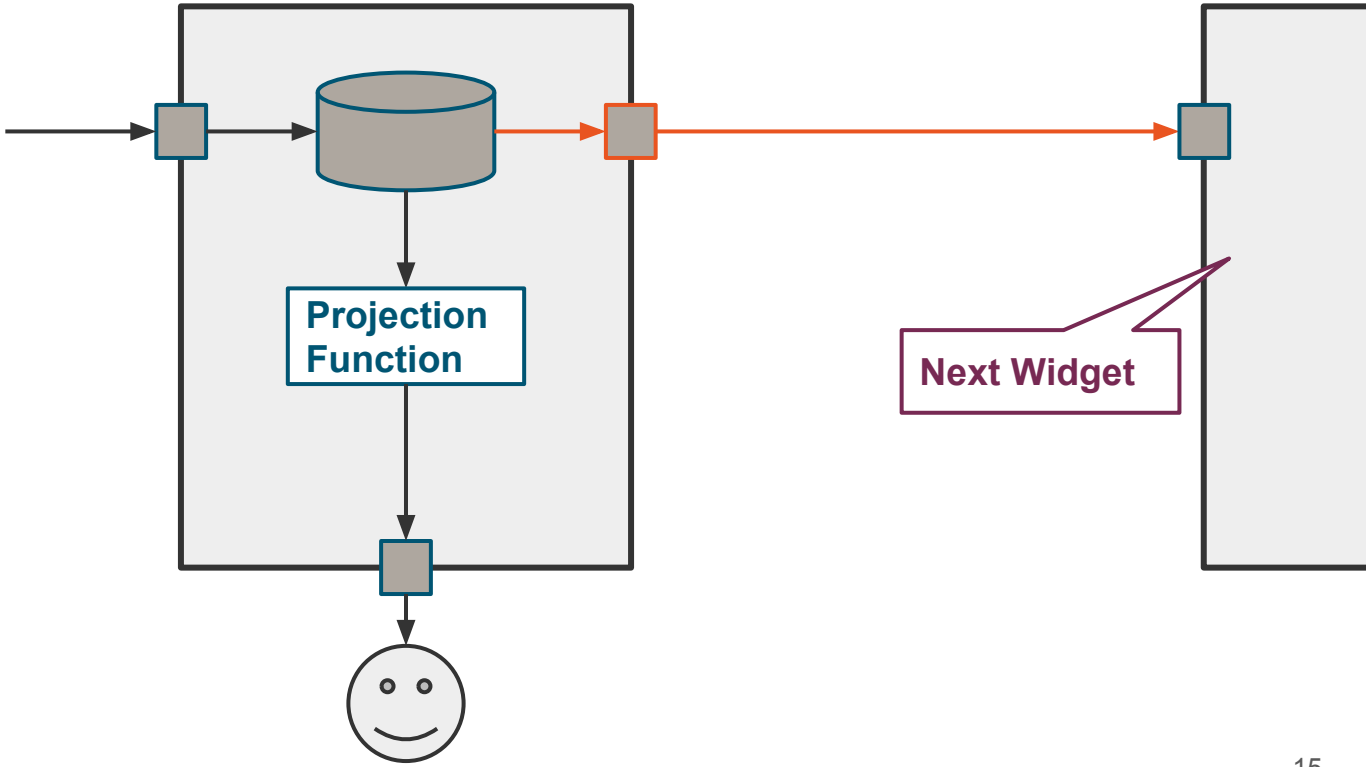



```
person => person.name
```

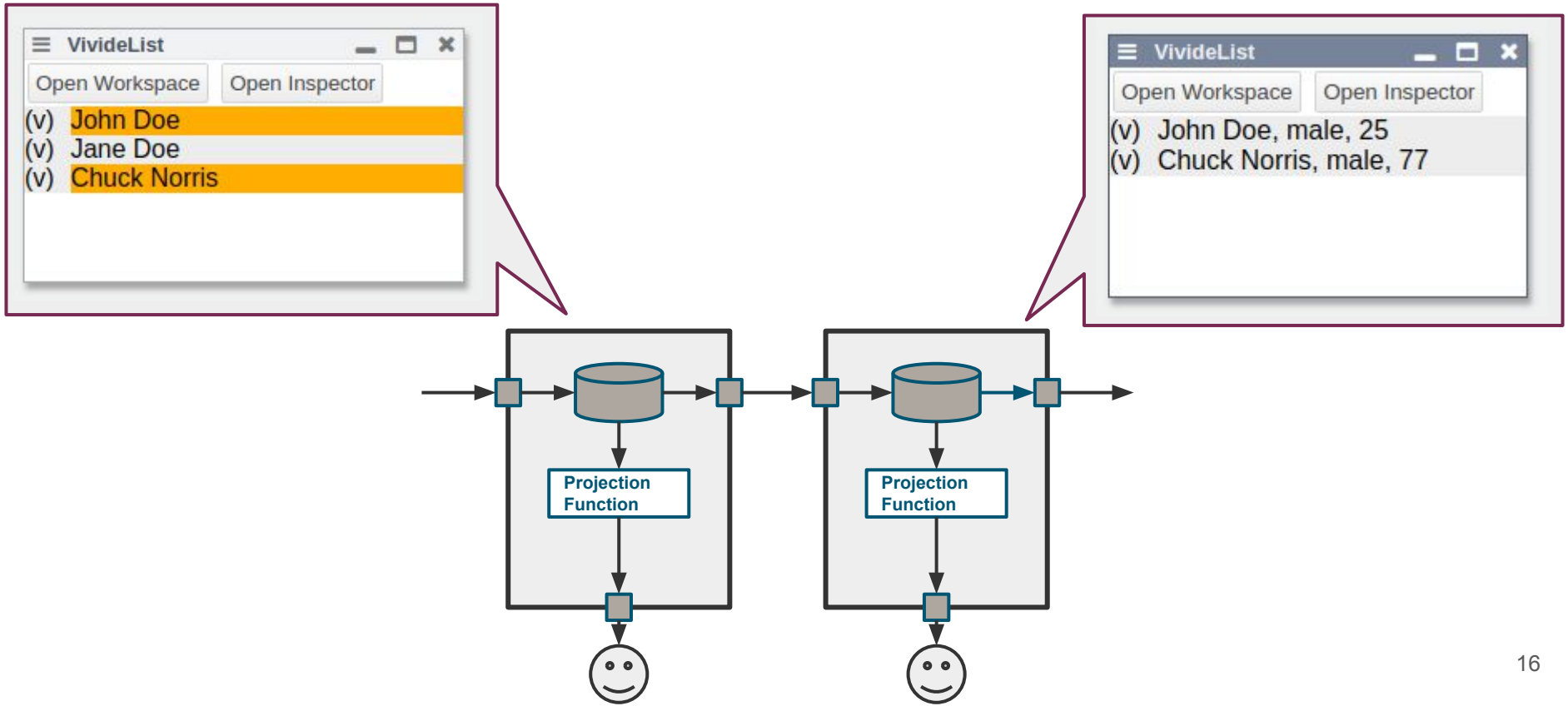
# Accessing Internal Data Representation



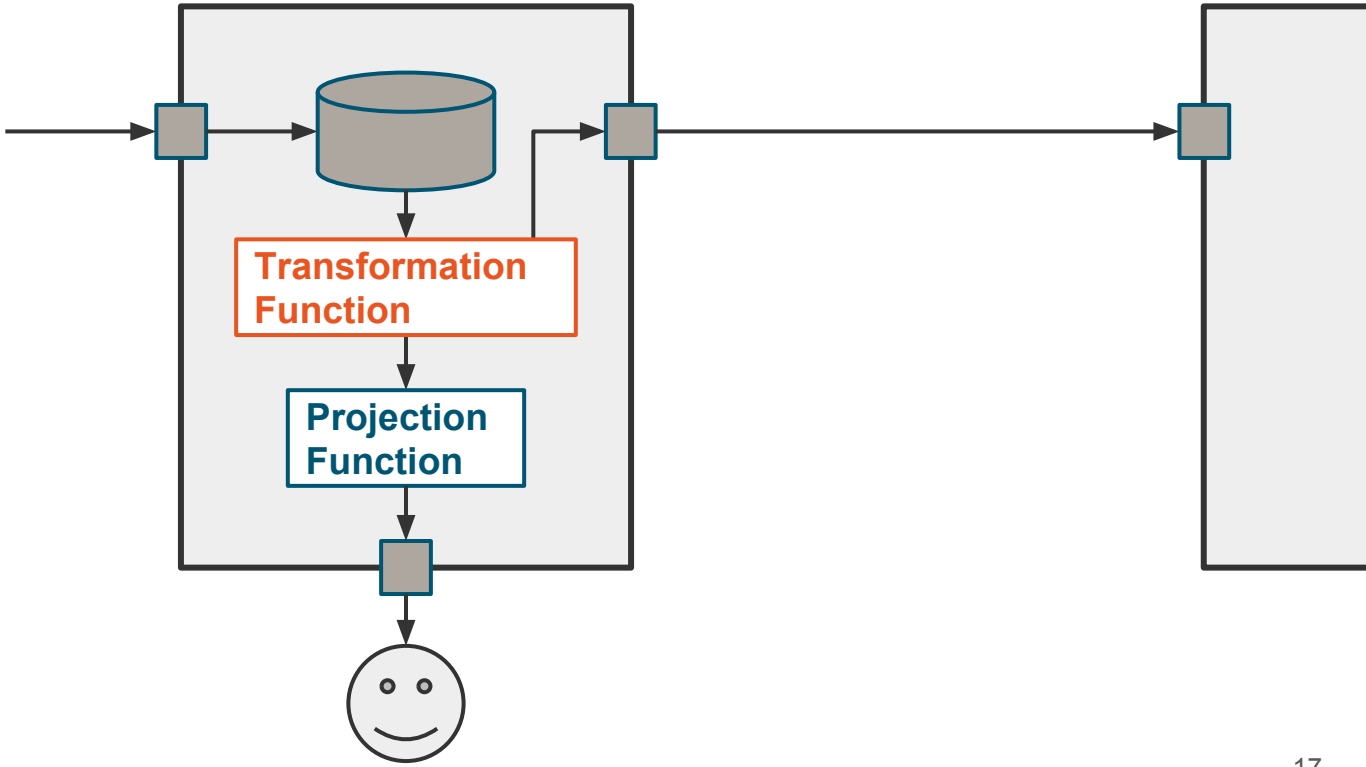
# Accessing Internal Data Representation



# Accessing Internal Data Representation



# Data Manipulation

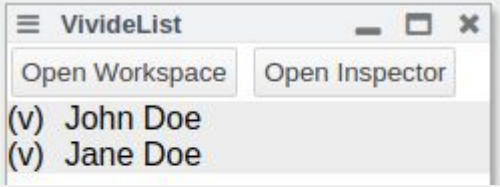


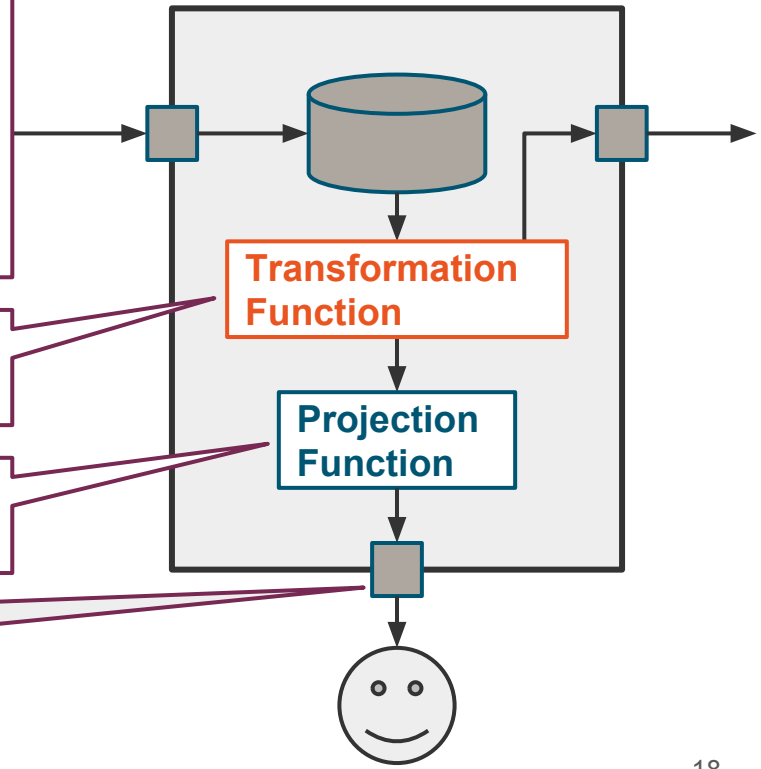
# Data Manipulation

```
[
 {name: "John Doe", age: 25, sex: "male", address: "New York"},
 {name: "Jane Doe", age: 24, sex: "female", address: "New York"},
 {name: "Chuck Norris", age: 77, sex: "male", address: "Everywhere"}
]
```

```
list => list.filter(person => person.age < 65)
```

```
person => person.name
```



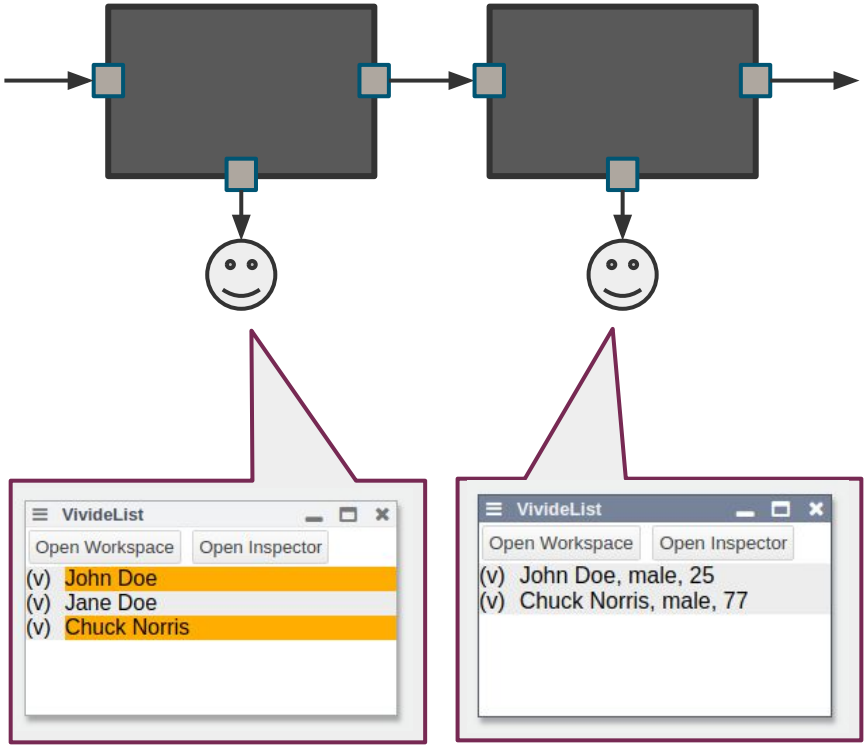
# Interaction between Widgets

# Interaction Design

If we connect widgets:
How should they interact?

**Which data** should be passed?

Design Choice:
Pass **selected data**



# Implementing Interaction

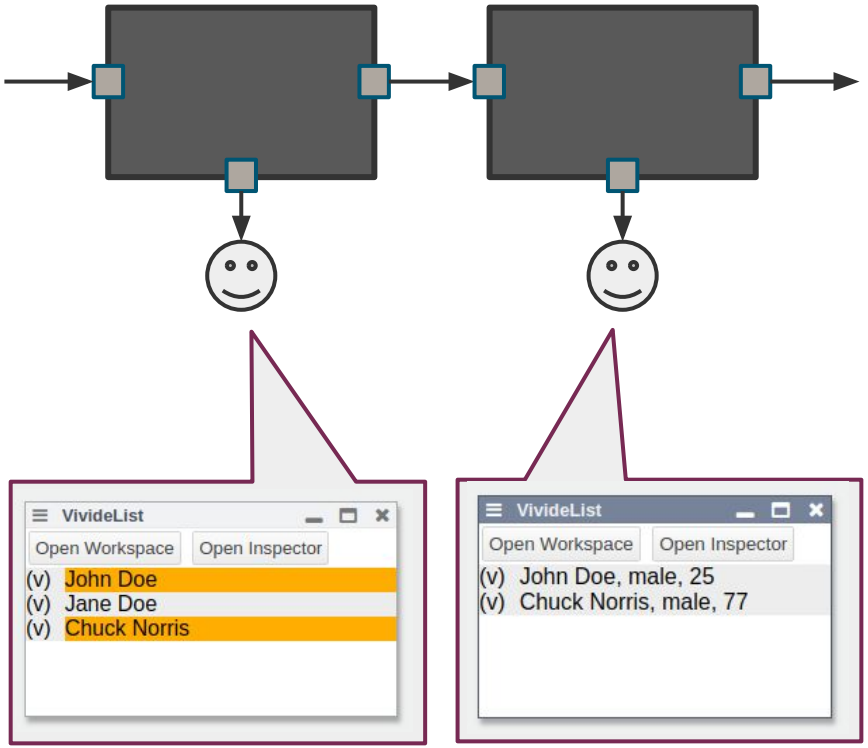
*Problem:*

How to propagate changes in selection?

We can not just pass data once.

*Solution:*

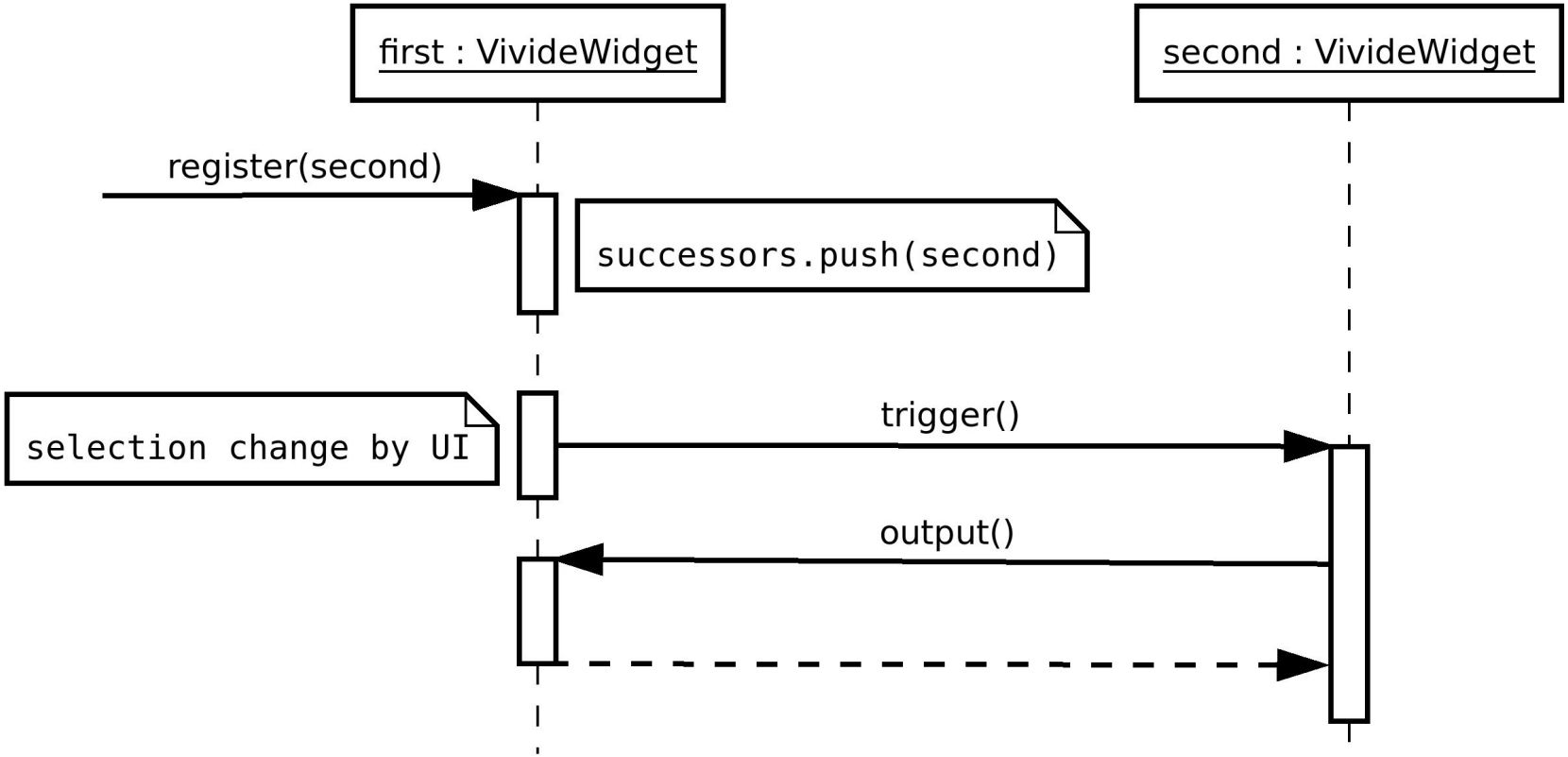
Observer Pattern




Each **Widget is Observer and Observable** at the same time.

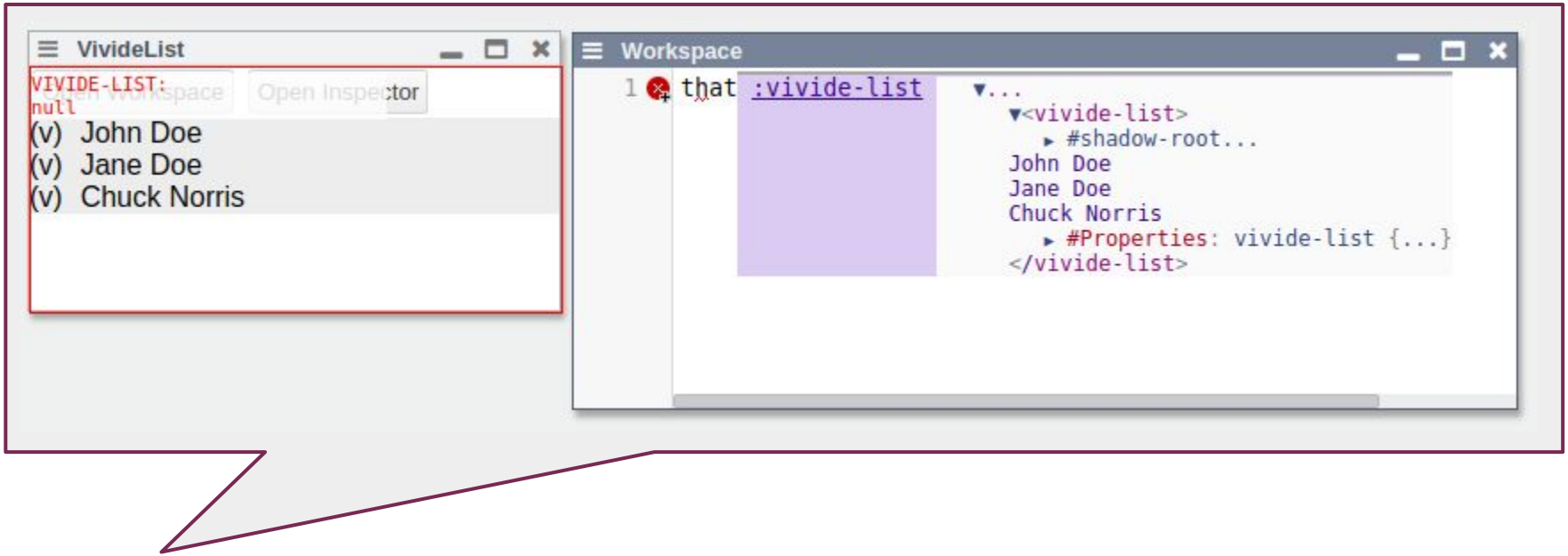
| **VivideWidget** |
| --- |
| `+predecessor`<br>`+successors` |
| `+register(anotherWidget)`<br>`+trigger()` |

# Implementing Interaction



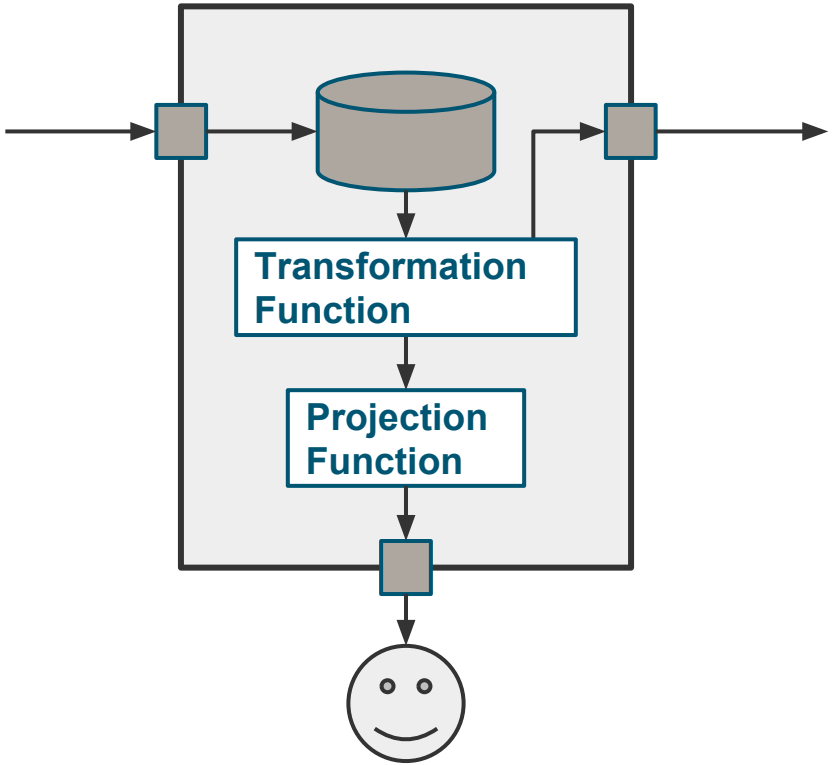
# Tooling

# Integration with Lively Tooling




"**Open Workspace**" button opens workspace with **that** bound to widget

# VivideInspector




Three **variable components**:

1. Internal Data
2. Transformation Function
3. Projection Function

# VivideInspector

VivideList

Open Workspace | Open Inspector

(v) John Doe
(v) Jane Doe
(v) Chuck Norris

VivideInspector

```
Model:
  ▼ : Array {
    ▼ 0: Object {
        address: "New York"
        age: 25
        name: "John Doe"
        ▼ orders: Array {
          ▼ 0: Object {
              item: "Pax"
              price: 210
          }
          ▼ 1: Object {
              item:
              "Komplement"
              price: 14
          }
        }
        sex: "male"
    }
    ▸ 1: Object {...}
    ▸ 2: Object {...}
  }
```

```
Transformation #0:
1  list => list

Projection #0:
1  person => person.name
```

Shows model, transformation, and projection

# Future Work

We have not covered (yet):

- Streamed data-access
- Multiple inputs for a single widget
- Recursively-defined infinite tree structures
- Graphical interaction (e.g. connecting by drag-and-drop)
- Polishing, Tooling, …

# Demo