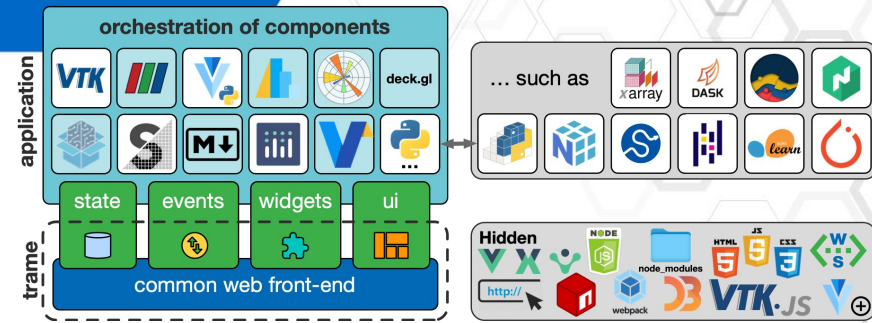# Trame - the basics

Sebastien Jourdain

kitware

What is trame?

- **Simple**

  All the logic and UI definition can be done in plain Python
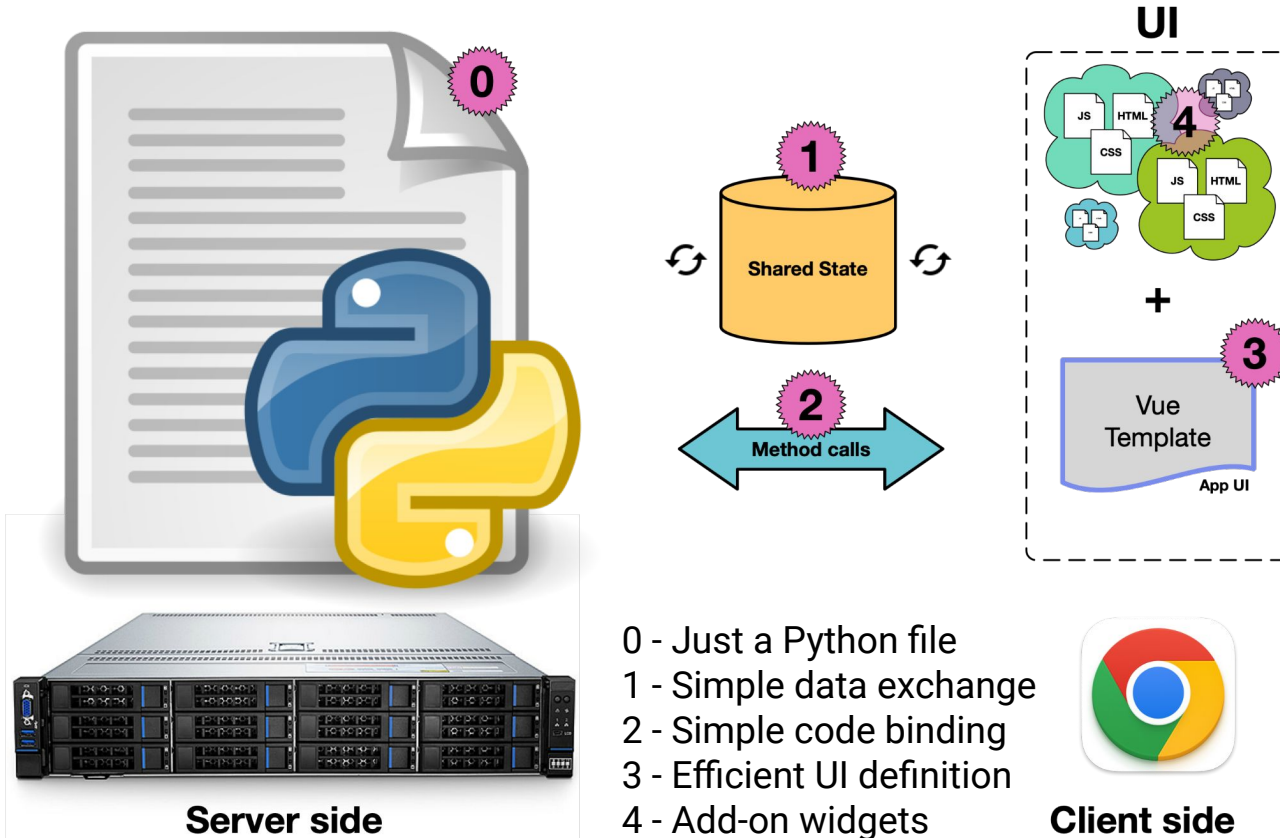
- **Powerful**

  Python offer scientific and information data visualization with capable data processing (numpy, Plotly, Matplotlib, VTK, ParaView…)

- **Ubiquitous**

  Runs on laptops, desktops, clusters, and the cloud while displaying everywhere (phone, tablet, laptop, workstation)
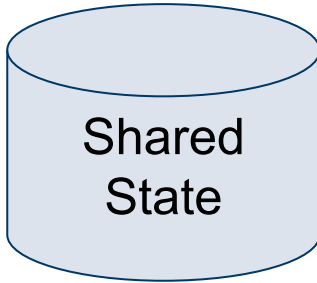
kitware

How trame works?

UI

0

1

Shared State

2

Method calls

4

JS
HTML
CSS

JS
HTML
CSS

+

3

Vue
Template

App UI

0 - Just a Python file
1 - Simple data exchange
2 - Simple code binding
3 - Efficient UI definition
4 - Add-on widgets

Server side

Client side

kitware

State handling

```python
from trame.app import get_server
server = get_server()
state = server.state
```

```python
@state.change("a", "b", "c")
def change_detected(a, b, c, d, **kwargs):
    pass


state.a = 10
print(f"a={state.a}")


def update_var(name, value):
    print(f"Before: {name}={state[name]}")
    state[name] = value
    print(f"After: {name}={state[name]}")
```

Shared
State

JS

```
Widget(key="welcome")              : Use string 'welcome'
Widget(key=("welcome",))           : Evaluate expression 'welcome'
Widget(key=("welcome","Hello"))    : Evaluate expression 'welcome' and initialize it to "Hello"
```

Kitware

Method handling and controller helper

```python
from trame.app import get_server
server = get_server()
ctrl = server.controller
```

```python
# @ctrl.set("hello_alias")
def hello(*args, **kwargs):
    print(args, kwargs)

@ctrl.add("hello_alias")
def hello2(*args, **kwargs):
    print("v2:", args, kwargs)


ctrl.hello_alias = hello
ctrl.hello_alias('arg_0', key='yes')
```

Method calls

JS

```
VBtn(click=fn)                                : Call the function with no args
VBtn(click=(fn, "[1, $event, 'hello']"))      : Call the function like fn(1, event, "hello")
VBtn(click=(fn, "[$event]", "{ x: 5 }"))      : Call the function like fn(event, x=5)
```

From JavaScript to Python

```html
<v-tooltip bottom>
  <template v-slot:activator="{ on, attrs }">
    <v-btn
      color="primary"
      dark
      v-bind="attrs"
      v-on="on"
    >
      Button
    </v-btn>
  </template>
  <span>Tooltip</span>
</v-tooltip>
```

@click=...

```python
from trame.widgets import html, vuetify

with vuetify.VTooltip(bottom=True) as debug:
    with vuetify.Template(v_slot_activator="{on, attrs}"):
        vuetify.VBtn(
            "Button",
            color="primary",
            dark=True,
            v_bind="attrs",
            v_on="on",
        )
    html.Span("Tooltip")

print(debug.html) # To validate that the template match
```

click=...

Any invalid Python character become "_" for any given attribute key.

https://vuetifyjs.com/en/components/tooltips/

kitware

Additional materials

- **Trame**
  - https://kitware.github.io/trame/

- **Course video**
  - https://vimeo.com/761096621/af2287747f

- **Course materials**
  - https://github.com/Kitware/trame-tutorial/tree/master/course/introduction

kitware