

cytoscape.js

Max Franz, Christian Tannus-
Lopes, Yue Dong

Outline

1. Problem space
2. What is cytoscape.js?
3. Why do we need it?
4. How is it implemented?
5. What can we do with it?

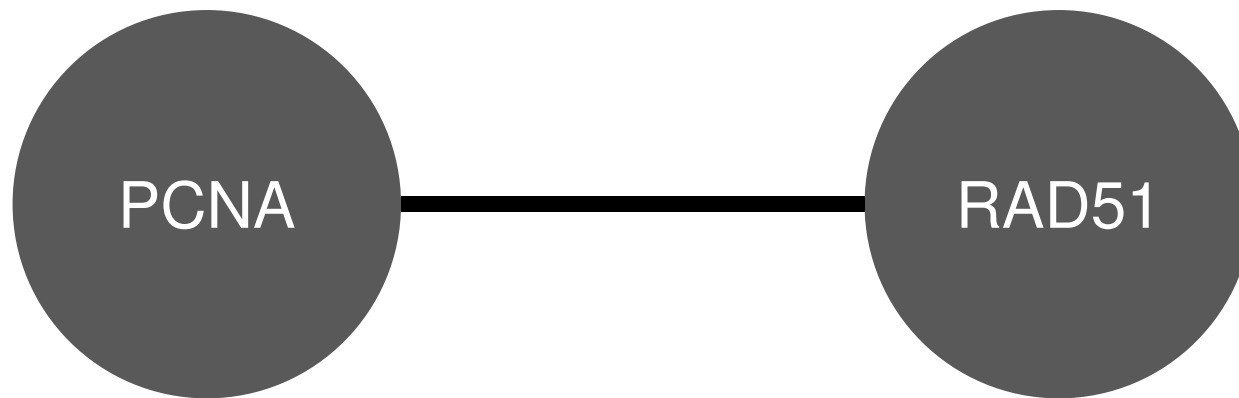
Outline

1. Problem space

2. What is cytoscape.js?
3. Why do we need it?
4. How is it implemented?
5. What can we do with it?

Entity-relationship data

e.g. PCNA interacts with RAD51



In general : graph data

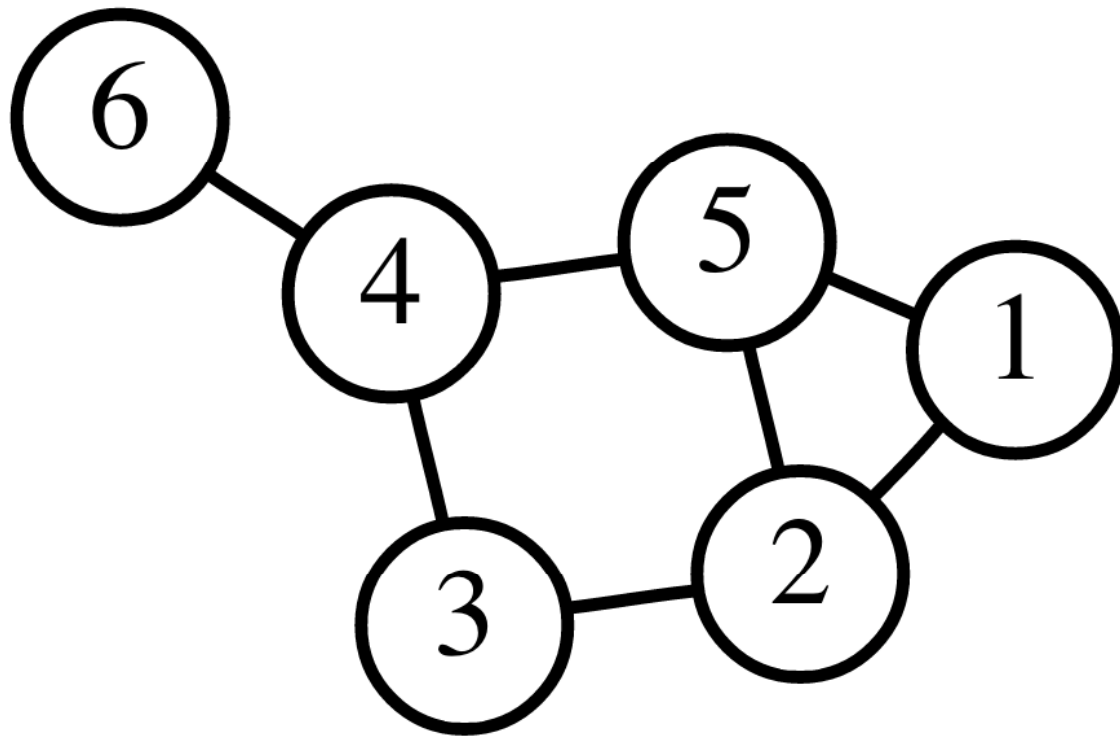


Image from Wikipedia ([http://en.wikipedia.org/wiki/Graph_\(mathematics\)](http://en.wikipedia.org/wiki/Graph_(mathematics)))

Outline

1. Problem space
- 2. What is cytoscape.js?**
3. Why do we need it?
4. How is it implemented?
5. What can we do with it?

cytoscape.js:

A JavaScript graph visualisation library that treats graph elements as though they were a part of the DOM

Previous version: Cytoscape Web



Then



HTML



Now

Outline

1. Problem space
2. What is cytoscape.js?
- 3. Why do we need it?**
4. How is it implemented?
5. What can we do with it?

HTML5 rendering technology



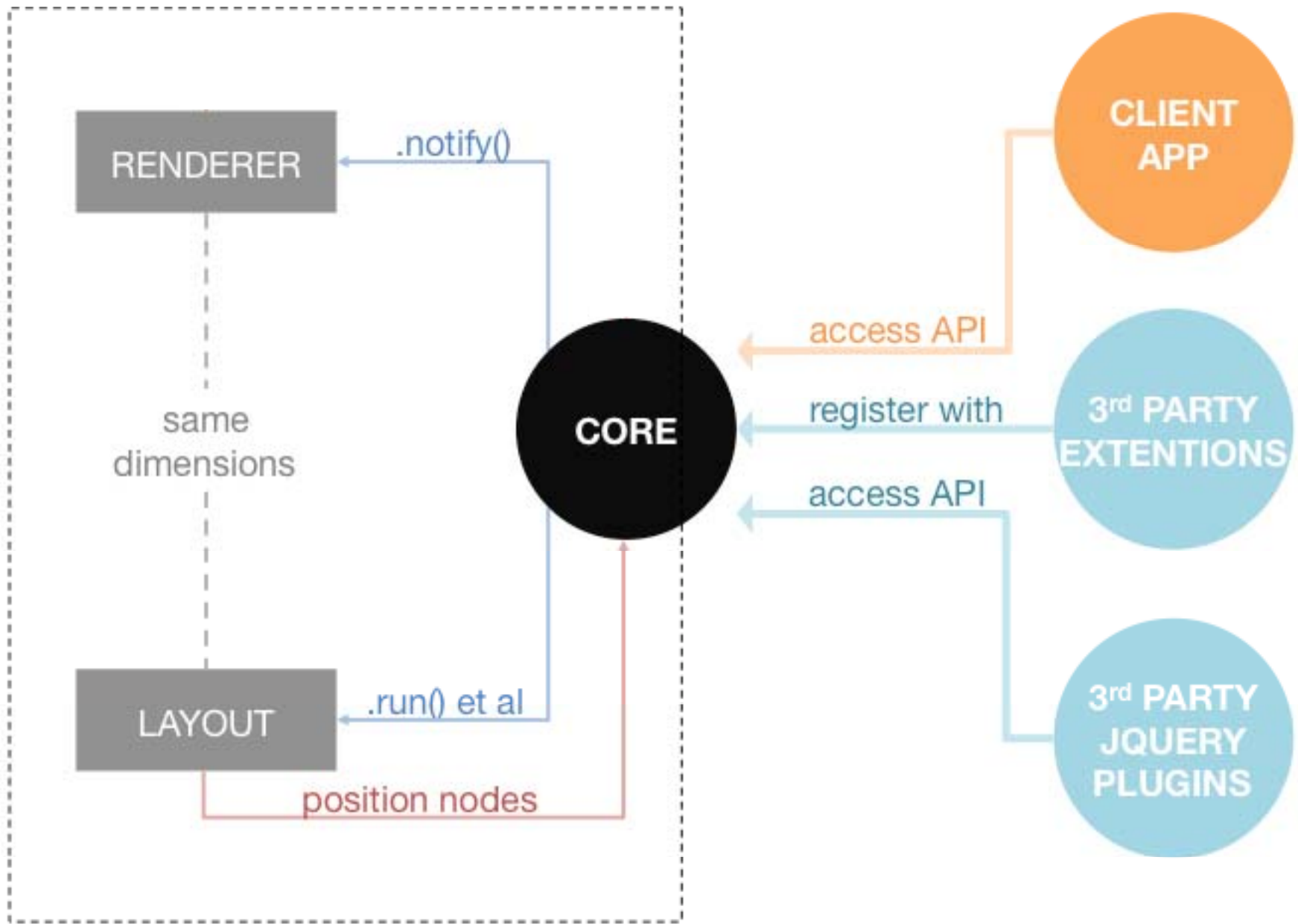
Other libraries are limited

- Style not configurable or is limited & programmatic
- Not dynamic
- Not interactive
- Not well suited for making web apps

Outline

1. Problem space
2. What is cytoscape.js?
3. Why do we need it?
- 4. How is it implemented?**
5. What can we do with it?

CYTOSCAPE.JS



Outline

1. Problem space
2. What is cytoscape.js?
3. Why do we need it?
4. How is it implemented?
- 5. What can we do with it?**

API

Inspired by



Selectors

cy.\$('node [weight > 50] {degree > 2}')

Curly for metadata

Square for client-defined data

Style

```
// Specify like CSS
cy.style()
  .selector('node')
  .css({
    'height': '20px',
    'width': '20px',
    'background-color': 'mapData(weight, 0,
      100, blue, red)'
  })
```

Separation of style and data

```
// Add and remove classes to change vis. state
```

```
cy.$('node[weight > 50]')
```

```
  .addClass('foo')
```

```
  .removeClass('bar')
```

```
// Manually set & override style
```

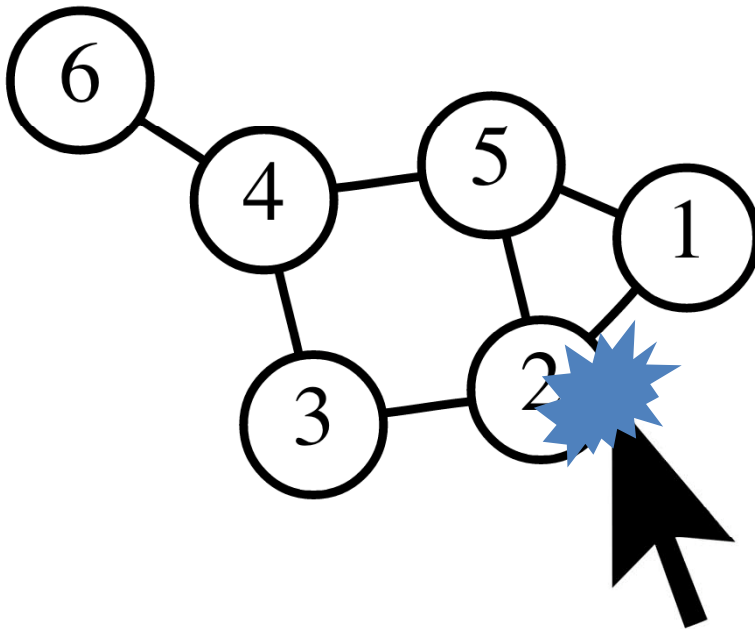
```
cy.$('node[weight > 50]')
```

```
  .css('background-color', 'blue')
```

Bind to & trigger events

```
cy.bind('click', 'node', function() {  
  console.log('clicked a node');  
});
```

```
cy.$('node[weight > 50]').trigger('click');
```



Graph analysis

- Parallel & codirected edges
- Intersections of the graph
- Calculate degree
- Find neighbourhood
- Filtering via selectors
- In future, more algorithms (walks, Djikstra, etc.)

Extensibility

- Layouts
- Renderers
- Functions
 - core `cy` instance object, e.g. `cy.foo()`
 - elements, e.g. `cy.$('node').bar()`

Not just a visualisation library

- Can be run headless, i.e. null renderer
- `cytoscape.js` + `node.js` = server side fun

Outline

1. Problem space
2. What is cytoscape.js?
3. Why do we need it?
4. How is it implemented?
5. What can we do with it?

Questions

<https://github.com/cytoscape/cytoscape.js>