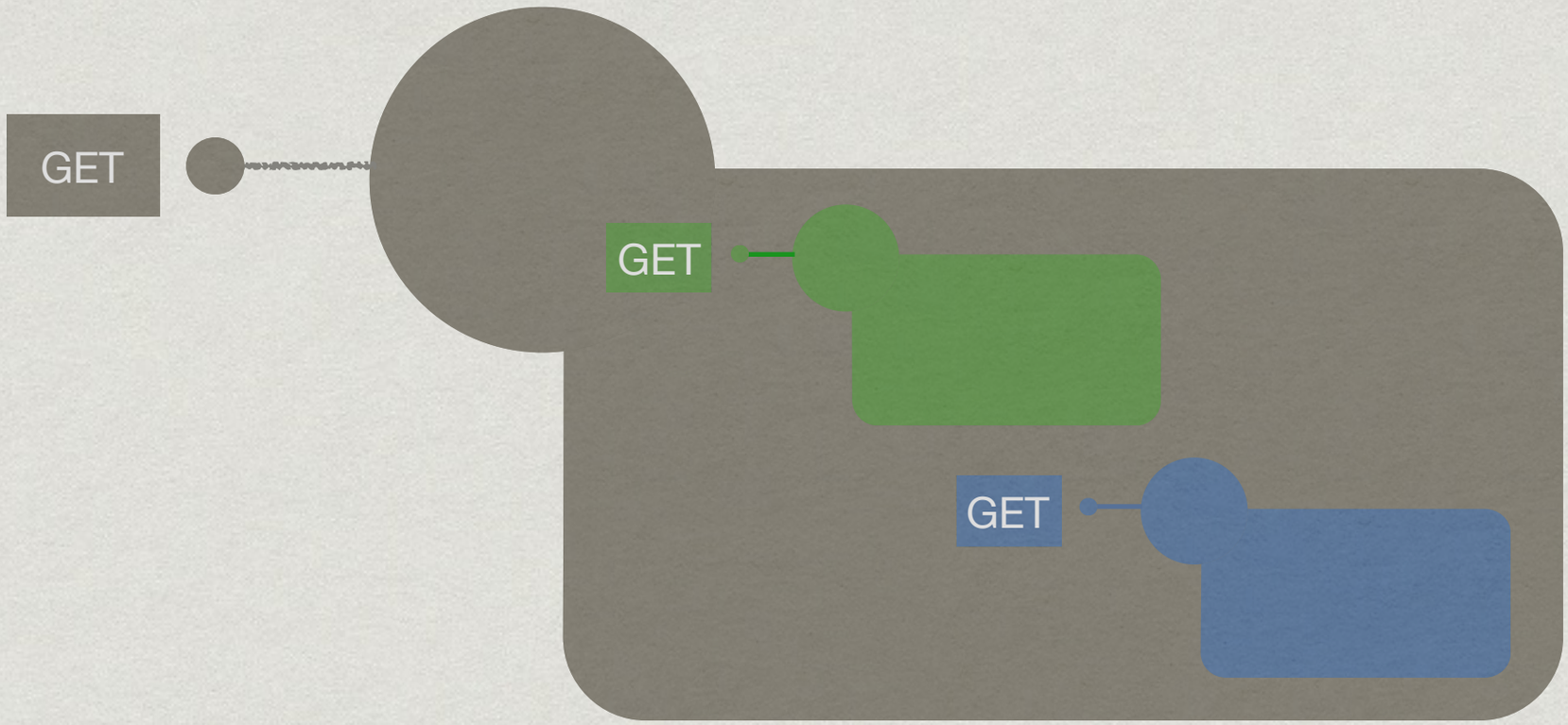
MASHUP AND COMPOSITION

Mashups

- Webapps aggregating information coming from different sources
- Directly in the browser



Few drawbacks

- Read-only
- Not reusable
- Single origin sandbox

Not reusable

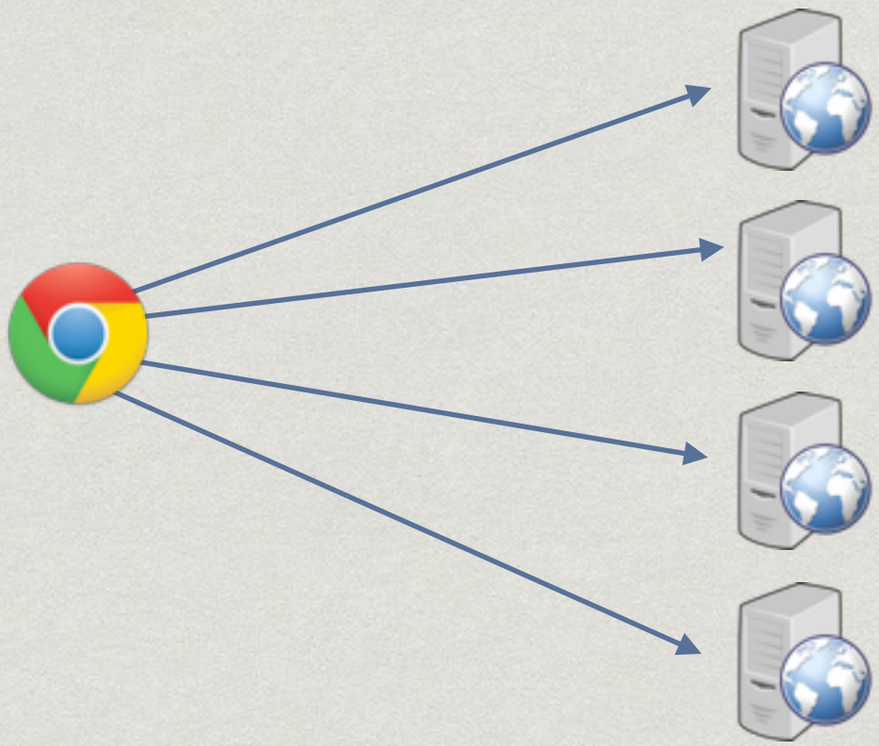
- Software composition is a well-known (and appreciated) technique
- It is about building blocks and composition operators
- If they are both sound by composition is recursive

Not reusable

- E.g,:
  - objects composed with references yield objects
  - composing tables with join yield a table
  - composing functions (FP & Math) yield a function
- Composing resources with mashup does not yield a resource

Single origin sandbox

- To avoid XSS modern browsers forbids a browser to contact many different servers



Summary

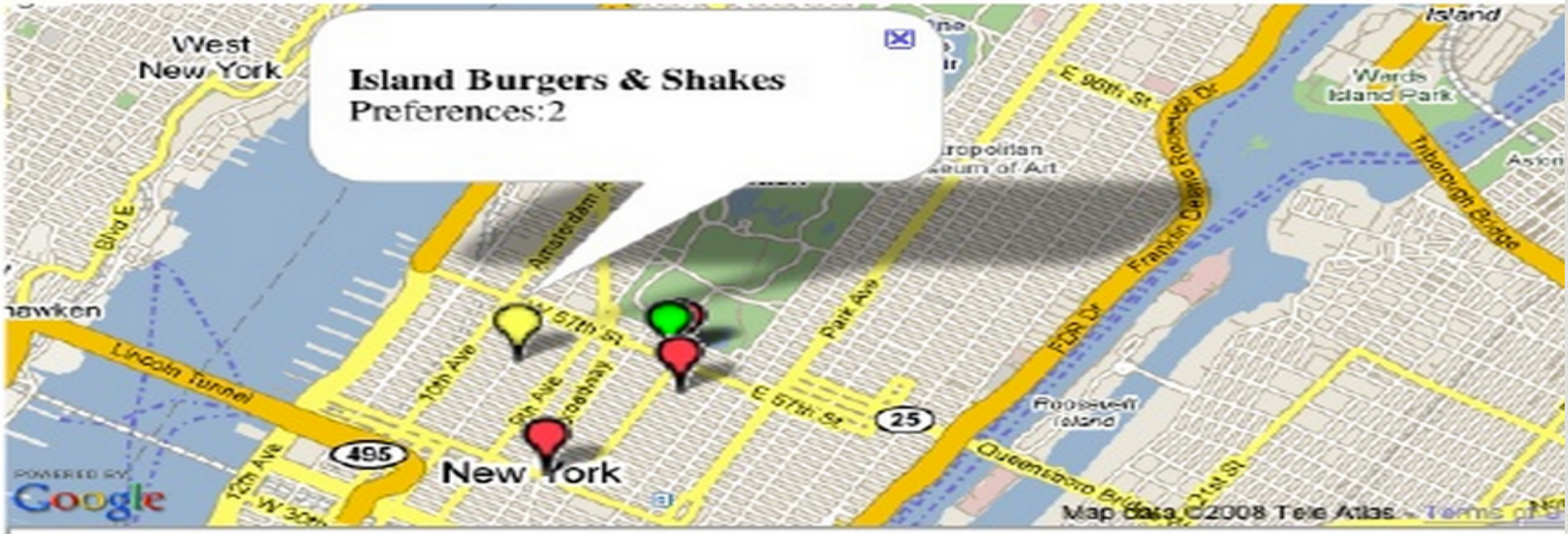
- We need to:
  - exec write operations
  - expose a (set of) new composite resources
  - serve it from one single origin

The DoodleMap example

- Geolocalized doodle
- Uses Yahoo local to search for places
- Visualizes them con Gmaps
- Allows for filling the doodle while looking at map
- Closes the poll when # participants reached

http://localhost:8080/kernel/

DoodleMap with JOpera



Poll: hamburger

CP has created this poll.

"10001"

	Gray's Papaya	Gray's Papaya	Hilton–Millenium	Island Burgers & Shakes	Le Parker Meridien–New York	Corner Bistro	ESPN Zone	Warwick New York Hotel	The Jekyll & Hyde Club	Dow the Hat
CP				OK		OK				
PA				OK	OK					

Component Services

- Doodle (D from now on):
  - POST (name, timeSlots, alternatives)—> new doodle
  - GET —> get current state
  - PUT (status=false)—> closes the poll

Component Services

- Gmaps (G from now on):
  - GET —> map with marker (addressability FTW!)
- Yahoo local (Y from now on):
  - GET —> places given the type (burgers), and the location

DoodleMap Resources

- /doodleMaps/
- /doodleMaps/{id}

/doodleMaps

- POST (name, #participants, location, timeSlots) :=

  alternatives = Y.GET(name, location)

  doodle = D.POST(name, alternatives, timeSlots)

- The new resource will contain:
  - doodle URL, #participants and alternatives

/doodleMaps/{id}

- GET := G.GET(alternatives) + D.GET(doodleUri)
- DELETE := D.DELETE(doodleUri)
- The resource will also poll the doodle (D.GET) to close it (D.PUT(closed)) if the #participants is reached

Summary

- Mashups and Resource composition are different
- And complementary