



adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 26-28 SEPTEMBER 2012

Sling Dynamic Include

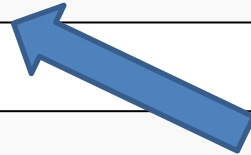
declarative caching of partially personalized pages

Tomasz Rękawek

- Our CQ site contains a lot of static pages
- Therefore they can be cached
- Unfortunately, there is one small dynamic component
 - Current user info
 - Latest news
 - Random article
 - External content
 - ...

Geometrixx example – userinfo component

CONTACT | (ADMINISTRATOR) MY PROFILE SIGN OUT



CONTACT | (ADMINISTRATOR) MY PROFILE SIGN OUT

Enter Query

GO

PRODUCTS SERVICES COMPANY EVENTS SUPPORT COMMUNITY

Visit our Booth at ShapeCon

G34 – 12/13 October in Las Vegas

Enter and win!

World Leader in Applied Geometry

GET THE GEOMETRIX APP

All things Geometrixx, anytime,

Problem

- Dispatcher is great in caching static pages
- However, even a small dynamic component makes it useless
- We want to cache the *almost* whole page
- Selected component should be included dynamically, directly from the CQ

Solution

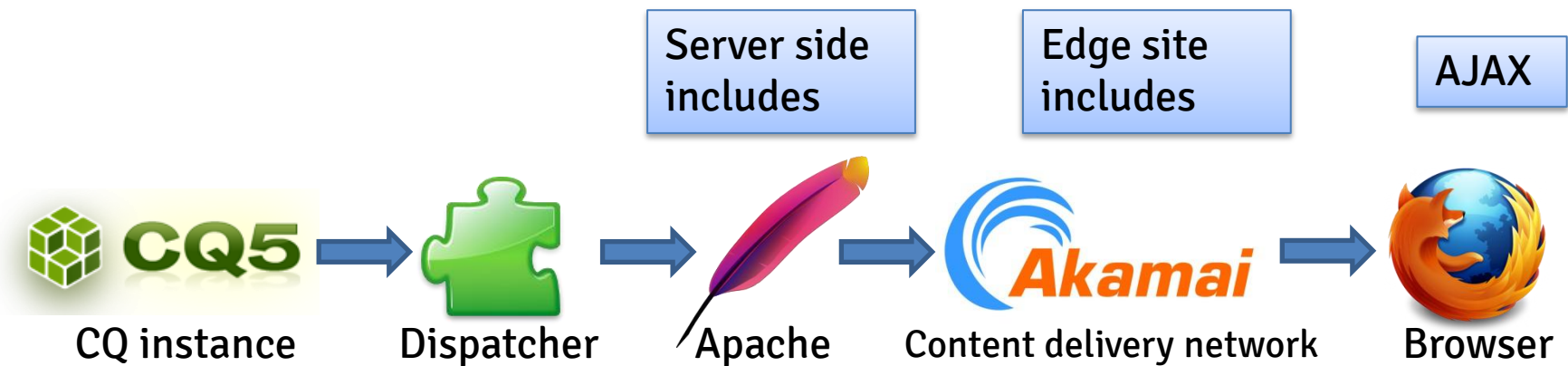
- Dispatcher should cache site with some static placeholder
- We replace placeholder with the dynamic content after dispatcher serves the site
- Helpful Sling feature: we can render only a part of the page using its URL

http://localhost:4503/content/geometrixx/en/_jcr_content/userinfo.html



Solution

- So, we can place some *include tag* which will be replaced with the component
- Include tag types available on different service layers between dispatcher and the client:



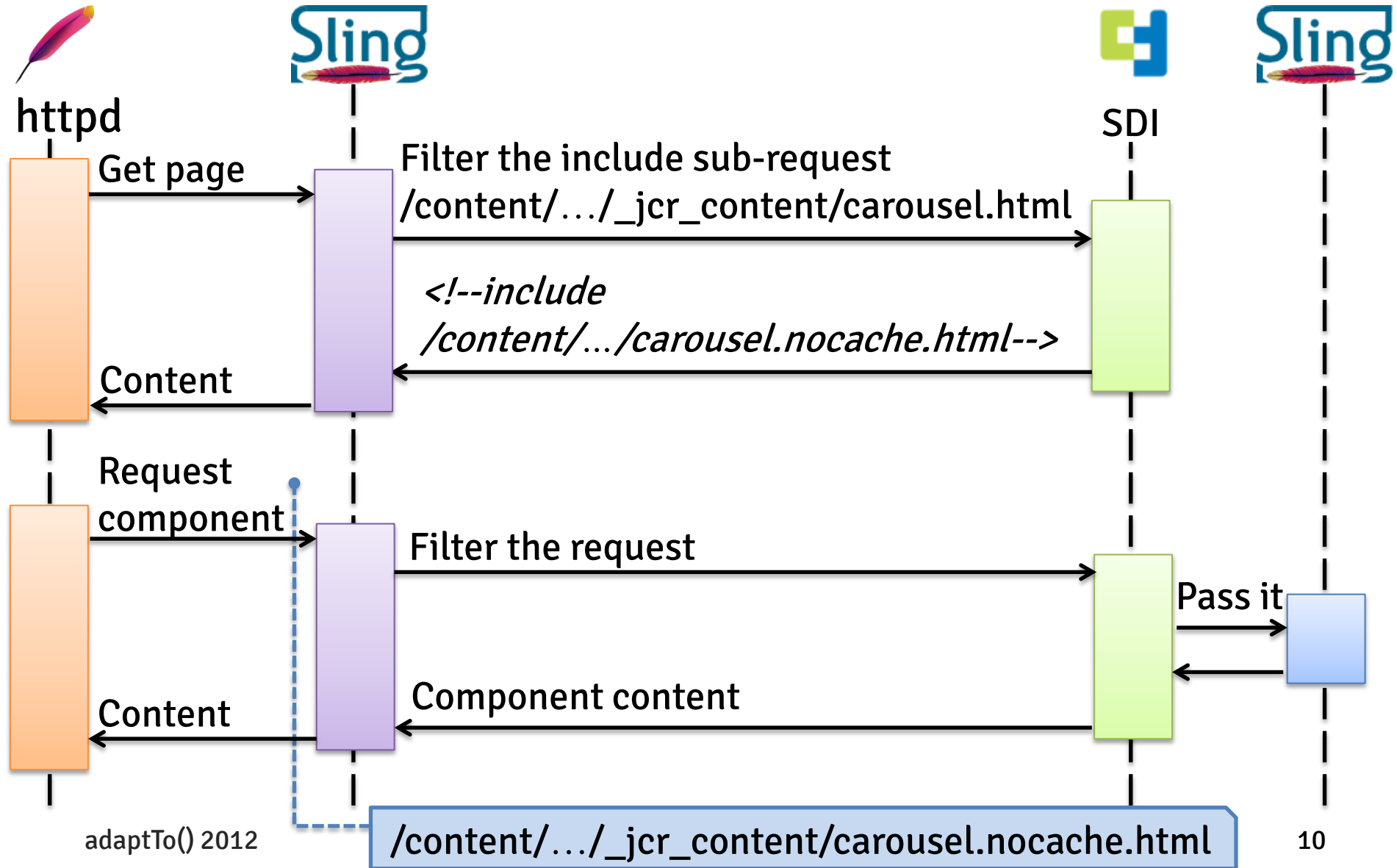
- **AJAX is a common solution here**
- **Drawbacks**
 - problems with components containing JS
 - may not work (smoothly or at all) on mobiles
 - not SEO friendly
 - slows down the site
 - flickering components

- **Advantages over AJAX**
 - Transparent for the browser / search engine
- **Drawbacks**
 - requires developers to have configured additional servers (Apache or Varnish)
 - developers should be aware of the environment
 - components should be written in a special way

Sling Dynamic Include

- Java servlet filter
- It affects components with specific resource types only
- Replaces it with:
 - SSI
 - ESI
 - AJAX include markup

SDI - how does it work?



SDI – advantages

- Approach transparent for the components
 - and developers as well
- Configured using OSGi console
 - can be disabled on the author / developers instances
 - include tag type can be easy changed
- Support for the nonexisting resources

Nonexisting resources

- Nonexisting (or synthetic) resource – resource without any JCR node

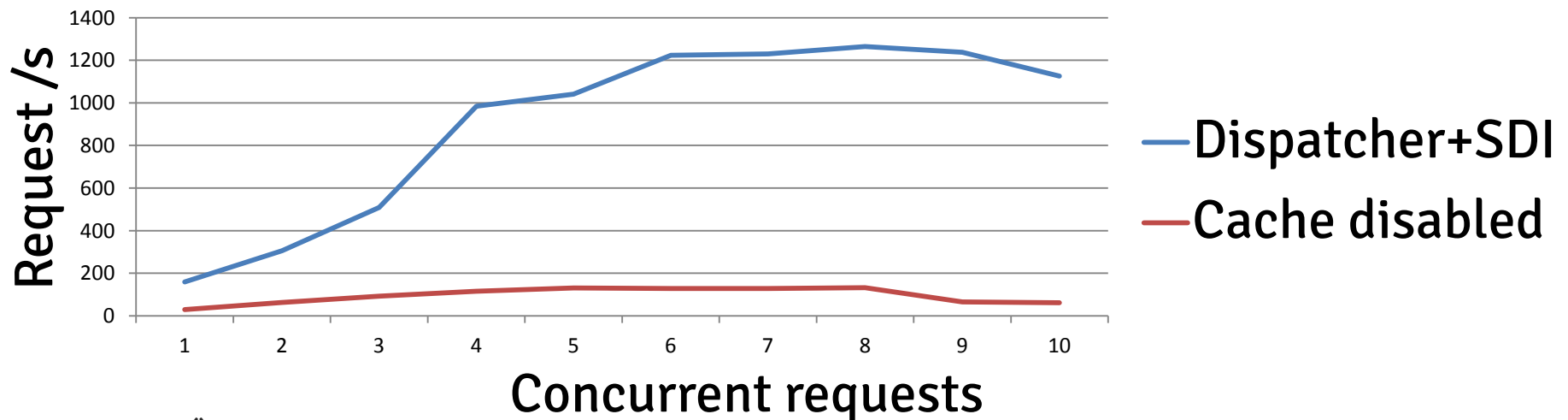
- JSP file excerpt:

```
<cq:include  
  path="footer"  
  resourceType="foundation/components/footer"/>
```

- But footer node doesn't exist
- Component is rendered normally, because it doesn't need any data
- However, requesting http://.../_jcr_content/footer.html will return 404
- SDI is able to recognize nonexisting resources and render them

Performance comparison

- Number of requests per second vs number of concurrent requests
- Geometrix site
- SDI includes *userinfo* component
- 8 threads: 1265 req/s with and 131 req/s without SDI
- Basically, it's a difference between rendering the whole page and a small part of it



- SDI is available on our Github
 - <https://github.com/Cognifide>
- Any questions?

