



FAST and EFFICIENT

Tizen HTML5 mobile applications

AKI SAARINEN, REAKTOR JAPAN

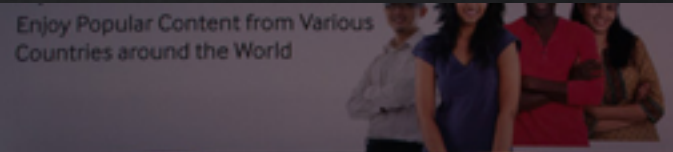
@akisaarinen

<http://reaktor.co.jp/en/>

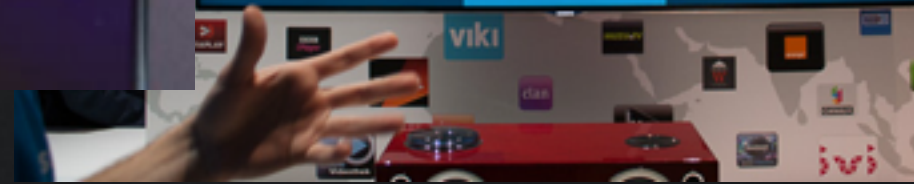
TIZEN™
DEVELOPER
CONFERENCE
2014
SAN FRANCISCO

FAST & EFFICIENT





Enjoy Popular Content from Various Countries around the World



- 1 Measure!
- 2 Start-up time
- 3 Run-time performance

1 MEASURE!

“

Measure before
optimizing

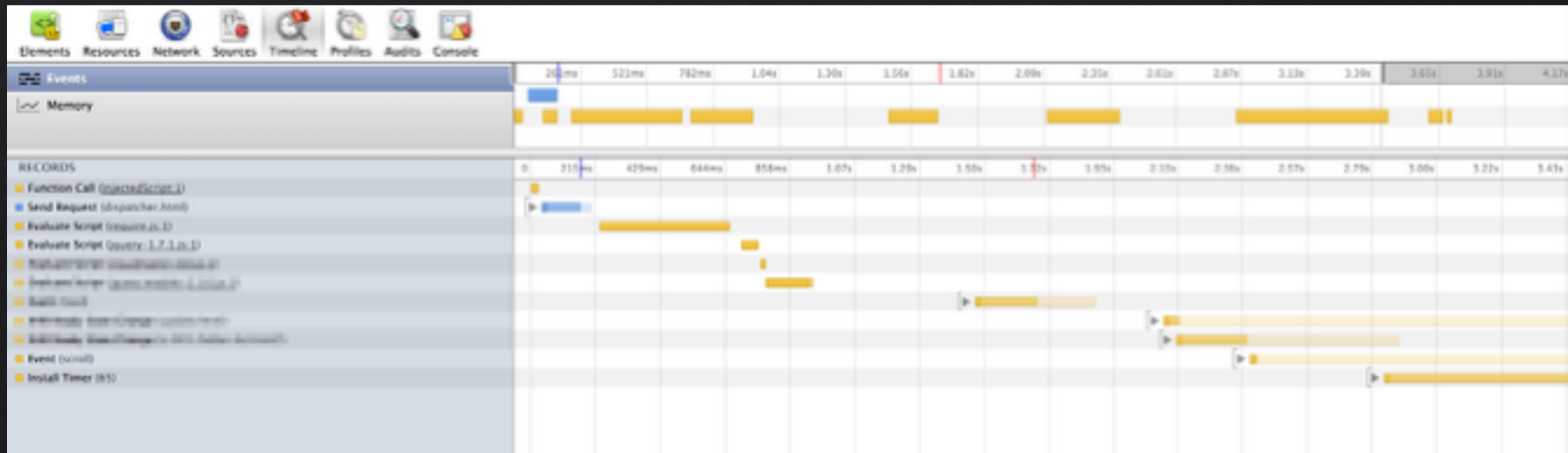
— a wise man



Available tools

- WebKit Web Inspector
- TizenDev: start-up
- TizenDev: framerate

WebKit Web Inspector



TizenDev

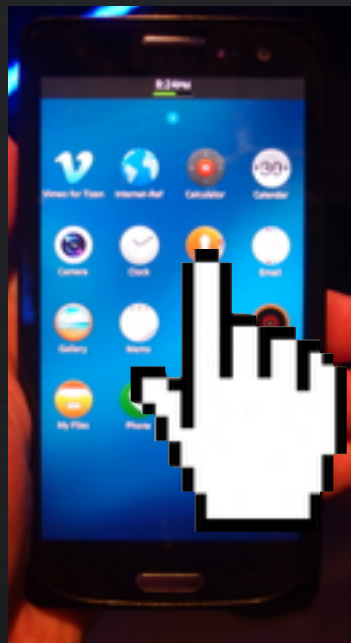
- Automated deploying of app
- Automated start-up timing
- Automated FPS measurements
- <http://github.com/reaktor/tizendev>

TizenDev: start-up time



runs : 30
mean : 1708ms
std : 63ms

TizenDev: framerate



samples:	100
mean:	58 FPS
std:	4 FPS

Available tools

- WebKit Web Inspector
- TizenDev: start-up
- TizenDev: framerate

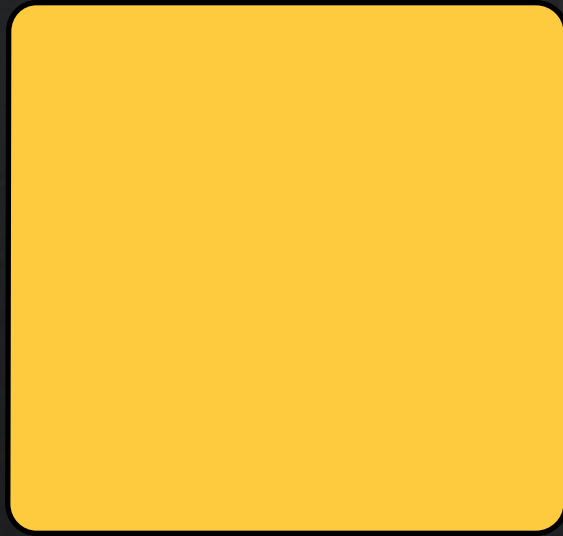
2 START-UP

LESS

IS MORE



- Lazy-loading
- Minification
- Reflow
- Native API calls
- Parallelization



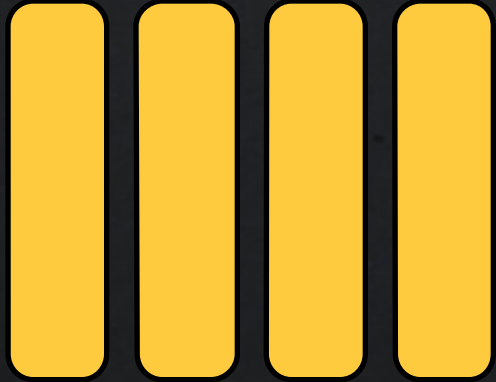
Large codebase,
all loaded and parsed
at start-up time



Large codebase,
all loaded and parsed
at start-up time



Code to show first screen



Modularized pieces
to show other views
on-demand

UglifyJS

1 kilobyte \approx 1 ms

Avoid reflow

Avoid reflow
REALLY!

Example:

Calling width() of an element

```
container.find("li").each(function() {  
    var listItem = $(this);  
    listItem.text(item.width());  
});
```



forces reflow

```
container.detach();
```

```
container.find("li").each(function() {  
    var listItem = $(this);  
    listItem.text(item.width());  
});
```

```
container.appendTo($("#body"));
```



```
container.detach();
```

```
container.find("li").each(function() {  
    var listItem = $(this);  
    listItem.text(item.width());  
});
```

```
container.appendTo($("#body"));
```

1000 elements (MacBook Pro)

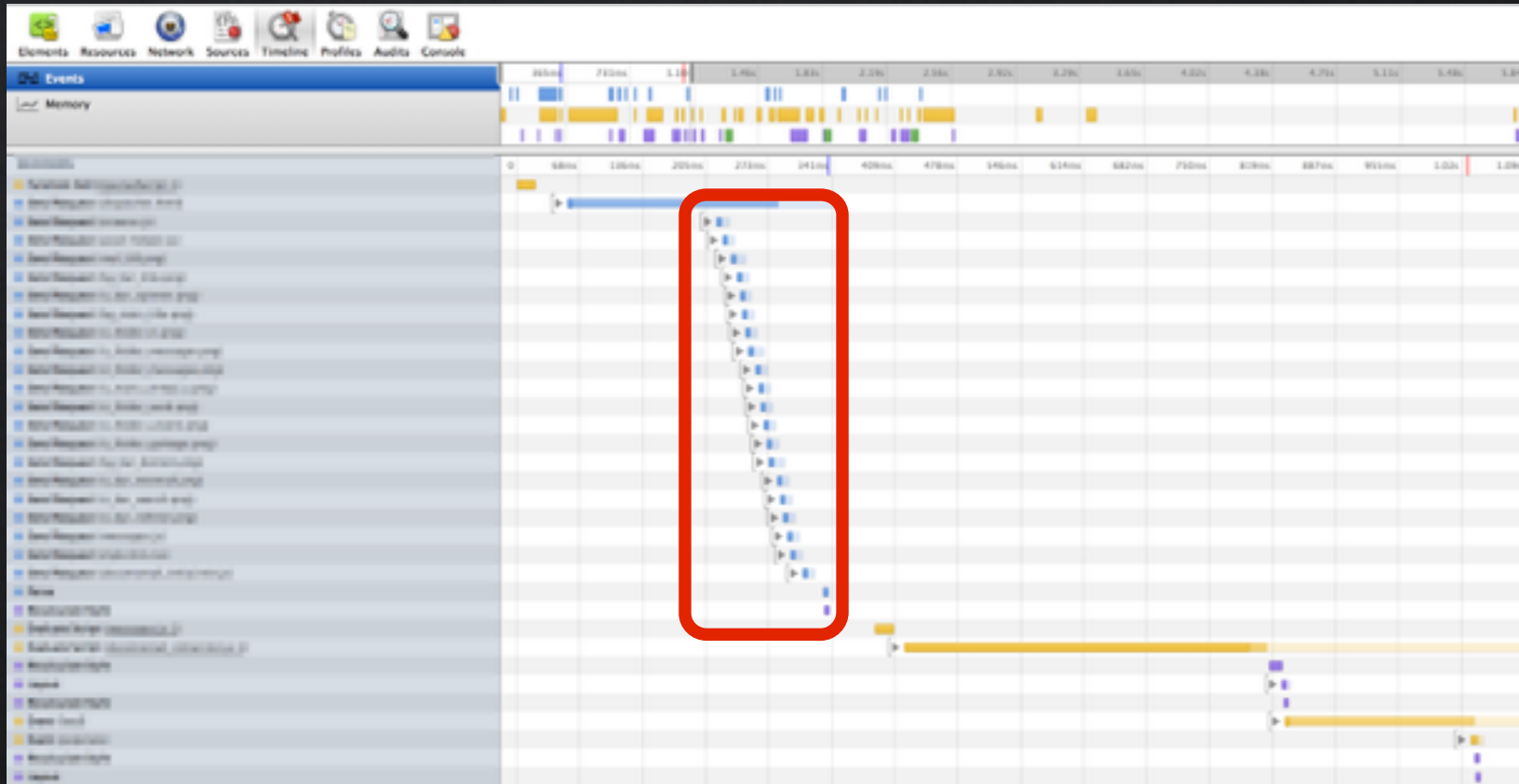
2000 ms → 60 ms

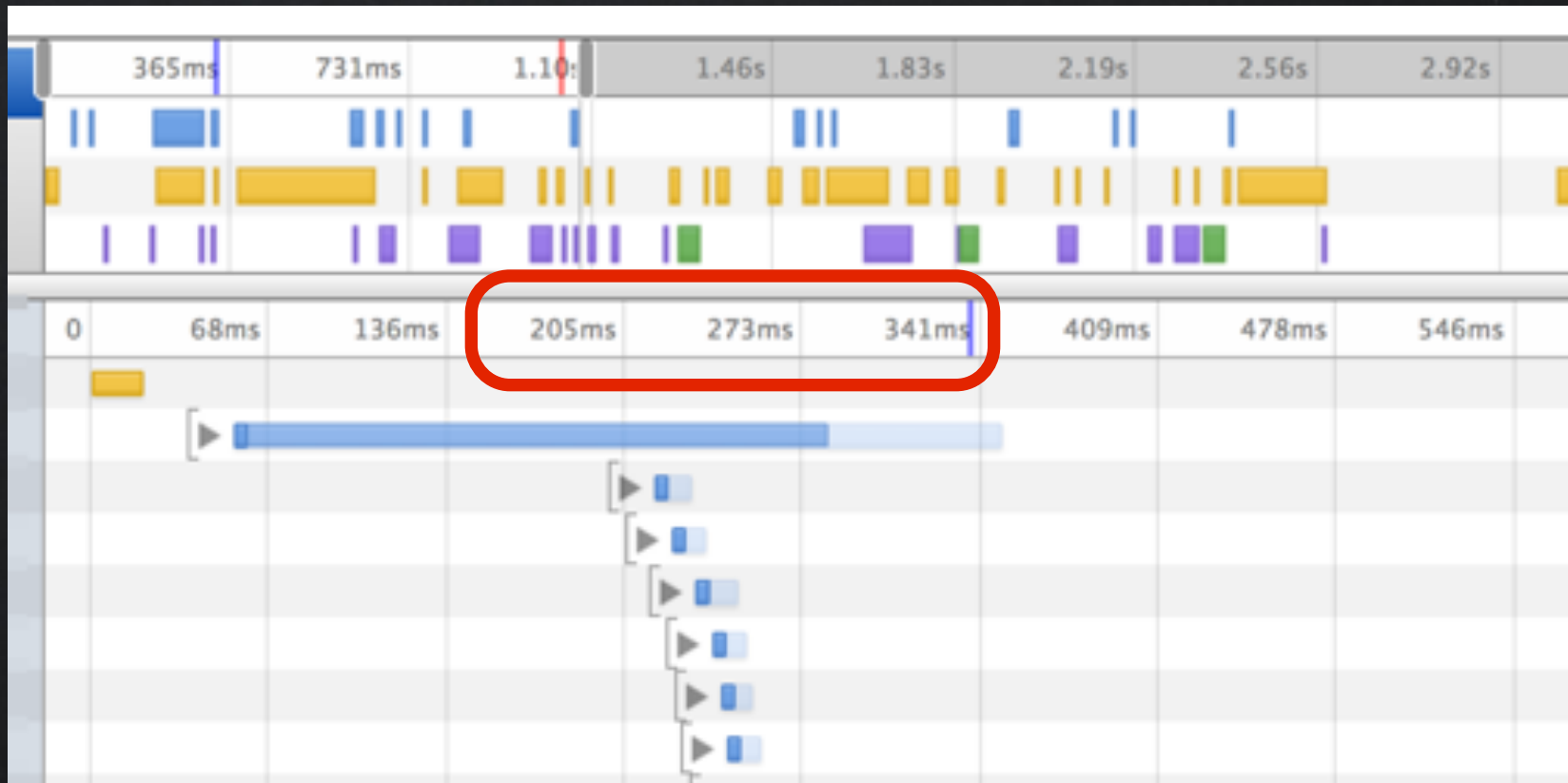
Native APIs

- Defer execution
- Use localstorage
- Only fetch needed data

Parallelize

- Resources
- Service calls





- Do lazy-loading
- Use minification
- Avoid reflow
- Careful with native APIs
- Parallelize

3 RUN-TIME



60 FPS

- DOM modifications
- Pre-loading
- CSS3 transitions
- Scrolling

DOM
=
SLOW

display: none;

+ 5-10 FPS



Accelerated CSS3 transitions

NO: `jQuery.animate()`

YES: `CSS3`

NO: left: 0px -> 100px

YES: translate3d()

NO: background-color: ...;

YES: opacity: 0.2;

Enable 3D acceleration

```
-webkit-transform: translate3d(0,0,0);
```

<http://stackoverflow.com/questions/3461441/prevent-flicker-on-webkit-transition-of-webkit-transform>

Trigger animation in next render cycle

```
setTimeout(function() {  
    element.css(  
        "-webkit-transform",  
        "translate3d(100,0,0)"  
    );  
}, 0);
```

Momentum scrolling

NO: iScroll or other JavaScript library

NO: overflow: scroll;

YES: -webkit-overflow-scroll: touch;

- DOM is slow
- Do pre-loading
- Use CSS3 transitions
- Use overflow scrolling

- 1 Measure!
- 2 Start-up time
- 3 Run-time performance

Re-cap!

- Performance is important
- Measure before optimizing
- Minimize actions at start-up
- Pay attention to FPS

Thank you!

@akisaarinen **Reaktor**



TIZEN™
DEVELOPER
CONFERENCE
2014
SAN FRANCISCO