

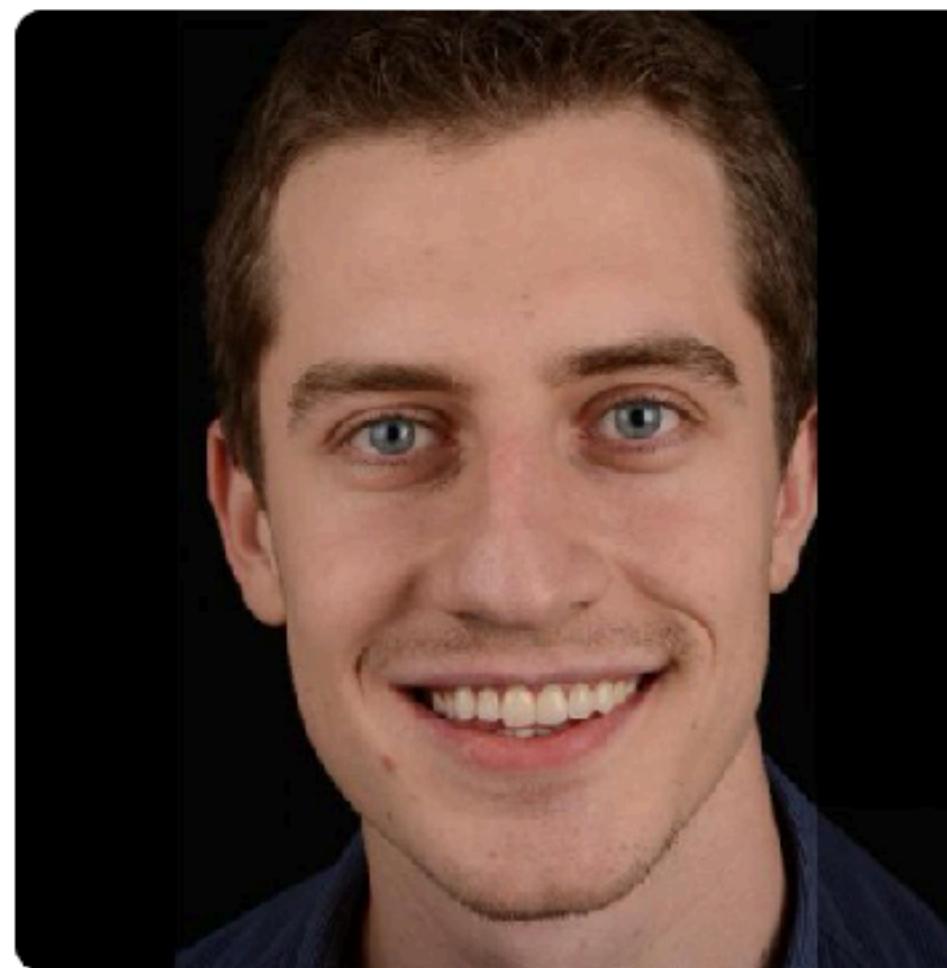


Faster Applications

Minko Gechev



rhyme

[Overview](#)

Repositories 210

Stars 99

Followers 3.6k

angular-style-guide

Set of best practices for Angular application development

★ 5k ⚡ 744

angular-seed

Modular starter project for Angular 2 (and beyond) with statically typed build and AoT compilation

● TypeScript ★ 4.4k ⚡ 1.7k

codelyzer

Linting for Angular projects.

● TypeScript ★ 1.4k ⚡ 133

angular/mobile-toolkit

JavaScript implementation of different computer science algorithms.

● TypeScript ★ 1,104 ⚡ 150

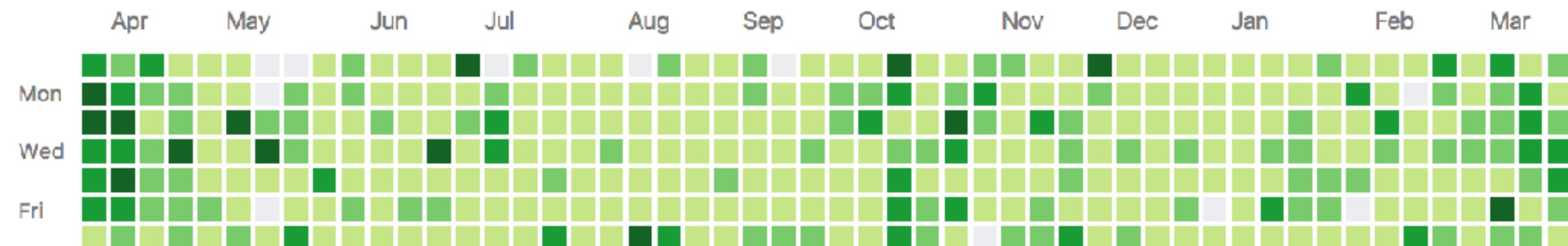
Minko Gechev
mgechev

Functional time traveler.

📍 California, USA

🔗 <http://blog.mgechev.com/>

Organizations



twitter.com/mgechev



github.com/mgechev



Minko Gechev

Foreword by Miško Hevery,
Creator of Angular

M
Ge

Switching to Angular

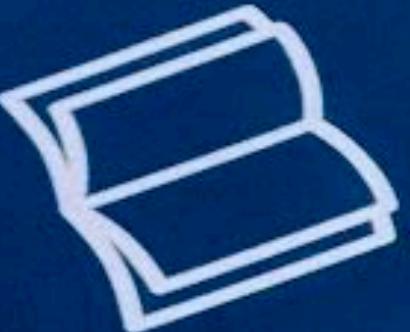
Third Edition

Foreword by Miško Hevery, Creator of AngularJS and Angular

Align with Google's long-term vision for Angular versions

Re

Minko Gechev



Minko

Runtime Performance



Simplified Business application



twitter.com/mgechev

A screenshot of a web application titled "Purely Fast" running on `localhost:5557`. The application features a clean, modern design with a light blue header and a white body. It is divided into two main sections: "Sales" on the left and "R&D" on the right. Each section includes a search bar at the top and a list of names below.

Sales Section:

- Search bar: Enter name here
- Items:

 - Cate (with number 75025 and delete icon)
 - Jeanna (with number 75025 and delete icon)
 - Lisbeth (with number 196418 and delete icon)
 - Neda (with number 75025 and delete icon)
 - Angel (with number 28657 and delete icon)
 - Correna (with number 28657 and delete icon)
 - Ingeborg (with number 75025 and delete icon)

R&D Section:

- Search bar: Enter name here
- Items:

 - Kimberley (with number 196418 and delete icon)
 - Marlie (with number 28657 and delete icon)
 - Marjorie (with number 75025 and delete icon)
 - Benetta (with number 121393 and delete icon)
 - Ruthann (with number 28657 and delete icon)
 - Sharyl (with number 196418 and delete icon)
 - Coriss (with number 196418 and delete icon)

A screenshot of a web browser window titled "Purely Fast" showing two lists of employees: "Sales" and "R&D".

The "Sales" list has the following data:

Sales			
Enter name here		Delete	
Nettle	196418		
Rosalie	317811		
Rhona	514229		
Talyah	196418		
Fancie	514229		
Kari	317811		
Caresa	514229		
Gerta	196418		
Modestine	196418		

The "R&D" list has the following data:

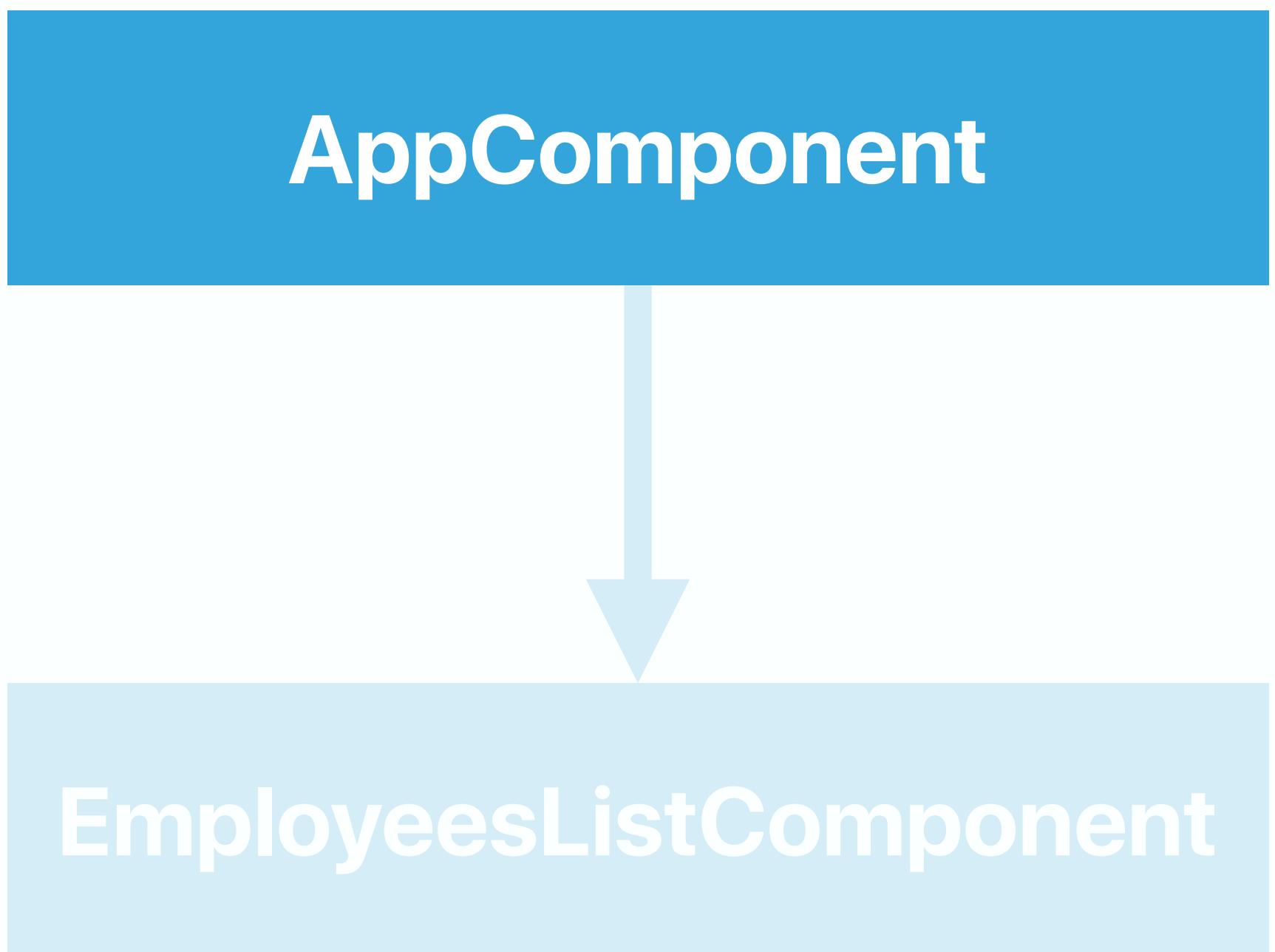
R&D			
Enter name here		Delete	
Sonja	317811		
Analiese	514229		
Concettina	196418		
Sherri	196418		
Emmalee	514229		
Darya	196418		
Alisun	317811		
Annamarie	514229		
Midge	514229		

AppComponent

EmployeesListComponent

The screenshot shows a web application interface with two main sections: "Sales" on the left and "R&D" on the right. Both sections have a header with a placeholder "Enter name here" and a list of employee names with their ID numbers in blue rounded rectangles and small trash can icons.

Section	Name	ID
Sales	Nettle	196418
Sales	Rosalie	317811
Sales	Rhona	514229
Sales	Talyah	196418
Sales	Fancie	514229
Sales	Kari	317811
Sales	Caresa	514229
Sales	Gerta	196418
Sales	Modestine	196418
R&D	Sonja	317811
R&D	Analiese	514229
R&D	Concettina	196418
R&D	Sherri	196418
R&D	Emmalee	514229
R&D	Darya	196418
R&D	Alisun	317811
R&D	Annamarie	514229
R&D	Midge	514229



Sales

Enter name here

Nettle	196418	Delete
Rosalie	317811	Delete
Rhona	514229	Delete
Talyah	196418	Delete
Fancie	514229	Delete
Kari	317811	Delete
Caresa	514229	Delete
Gerta	196418	Delete
Modestine	196418	Delete

R&D

Enter name here

Sonja	317811	Delete
Analiese	514229	Delete
Concettina	196418	Delete
Sherri	196418	Delete
Emmalee	514229	Delete
Darya	196418	Delete
Alisun	317811	Delete
Annamarie	514229	Delete
Midge	514229	Delete

AppComponent

EmployeesListComponent

Sales

Enter name here

Jeanna

75025



Carla

196418



Ezmeralda

121393



Karena

196418



Birdie

28657



Angy

75025



Rennie

121393



Lidia

46368



```
<input [(ngModel)]="label"
(keydown)="handleKey($event)">

<mat-list-item
*ngFor="let item of data">
  {{ item.label }}
  {{ calculate(item.num) }}
</mat-list-item>
```

Sales

Enter name here

Jeanna

75025



Carla

196418



Ezmeralda

121393



Karena

196418



Birdie

28657



Angy

75025



Rennie

121393



Lidia

46368



```
<input [(ngModel)]="label"  
        (keydown)="handleKey($event)">
```

```
<mat-list-item  
  *ngFor="let item of data">
```

```
  {{ item.label }}
```

```
  {{ calculate(item.num) }}
```

```
</mat-list-item>
```

Sales

Enter name here

Jeanna

75025



Carla

196418



Ezmeralda

121393



Karena

196418



Birdie

28657



Angy

75025



Rennie

121393



Lidia

46368



```
<input [(ngModel)]="label"
  (keydown)="handleKey($event)">

<mat-list-item
  *ngFor="let item of data">
  {{ item.label }}
  {{ calculate(item.num) }}
</mat-list-item>
```

Sales

Enter name here

Jeanna

75025



Carla

196418



Ezmeralda

121393



Karena

196418



Birdie

28657



Angy

75025



Rennie

121393



Lidia

46368



```
<input [(ngModel)]="label"  
        (keydown)="handleKey($event)">
```

```
<mat-list-item  
  *ngFor="let item of data">
```

```
  {{ item.label }}
```

```
  {{ calculate(item.num) }}
```

```
</mat-list-item>
```

```
@Component(...)
export class EmployeeListComponent {
  @Input() data: EmployeeData[];
  @Output() remove = new EventEmitter<EmployeeData>();
  @Output() add = new EventEmitter<string>();

  handleKey(event: any) { ... }

  calculate(num: number) {
    return fibonacci(num);
  }
}
```

```
@Component(...)
export class EmployeeListComponent {
  @Input() data: EmployeeData[];
  @Output() remove = new EventEmitter<EmployeeData>();
  @Output() add = new EventEmitter<string>();

  handleKey(event: any) { ... }

  calculate(num: number) {
    return fibonacci(num);
  }
}
```

AppComponent

data

EmployeesListComponent

```
@Component(...)
export class EmployeeListComponent {
  @Input() data: EmployeeData[];
  @Output() remove = new EventEmitter<EmployeeData>();
  @Output() add = new EventEmitter<string>();

  handleKey(event: any) { ... }

  calculate(num: number) {
    return fibonacci(num);
  }
}
```

Sales

Enter name here

Jeanna

75025



Carla

196418



Ezmeralda

121393



Karena

196418



Birdie

28657



Angy

75025



Rennie

121393

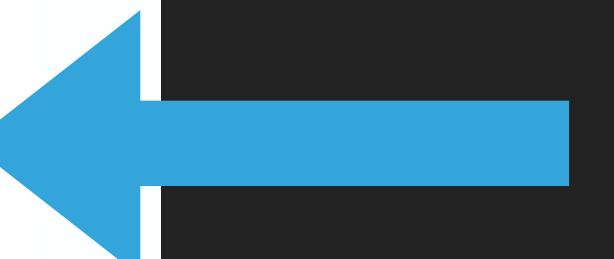


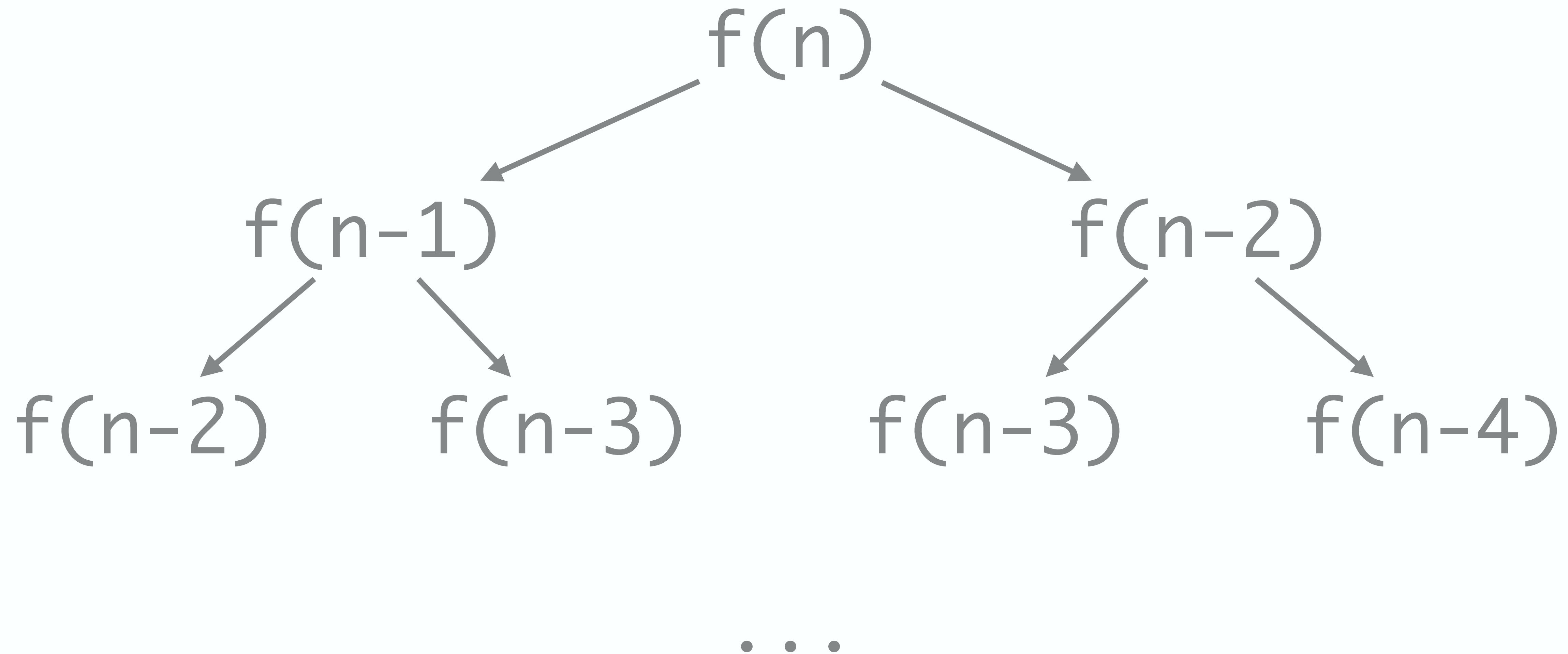
Lidia

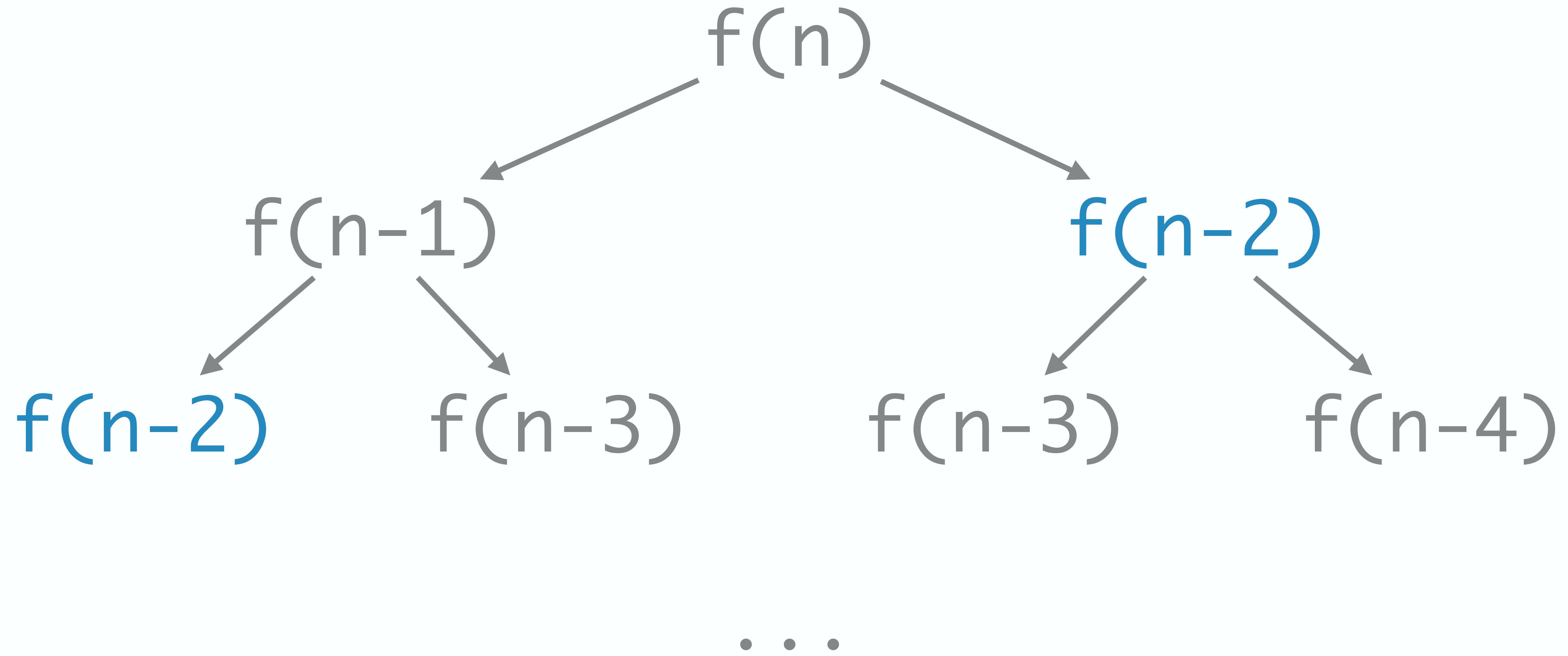
46368

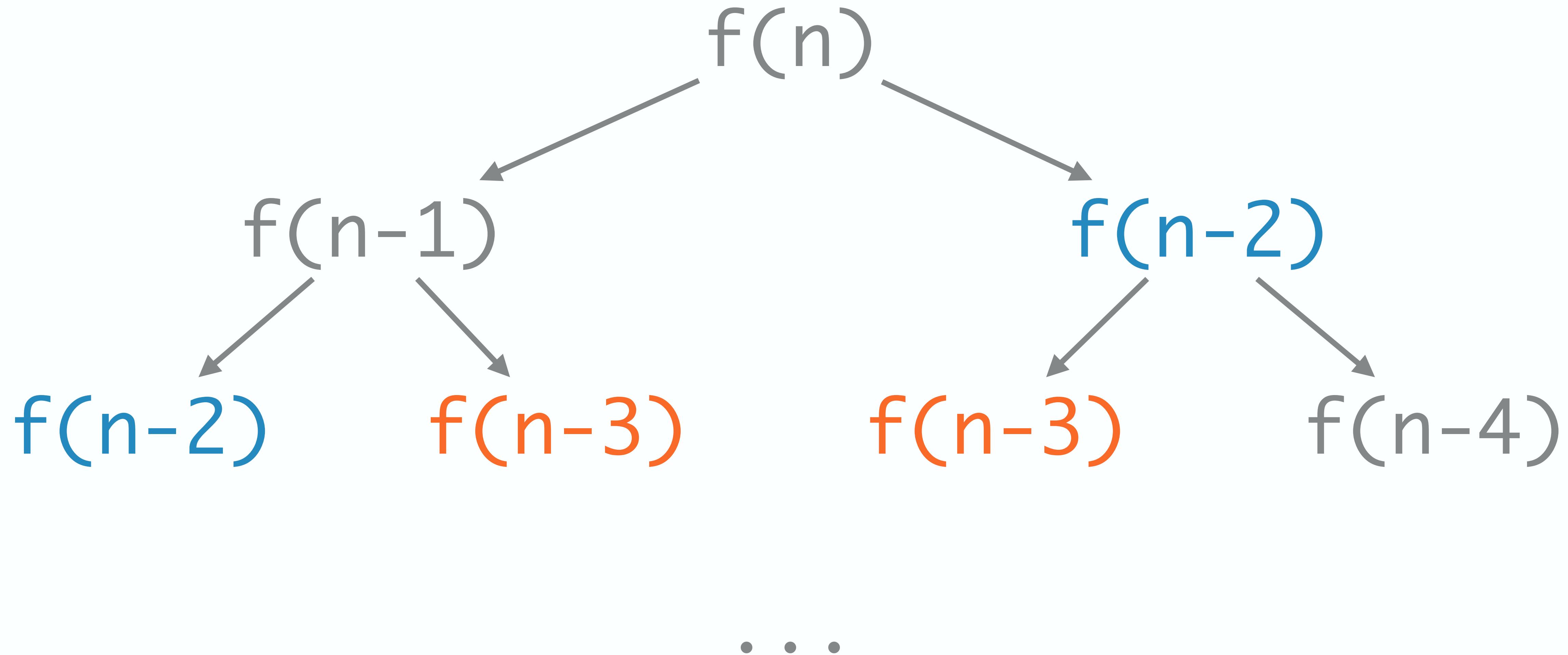


```
const fibonacci = n => {
  if (n === 1 || n === 2)
    return 1;
  return fibonacci(n - 1)
    + fibonacci(n - 2);
};
```

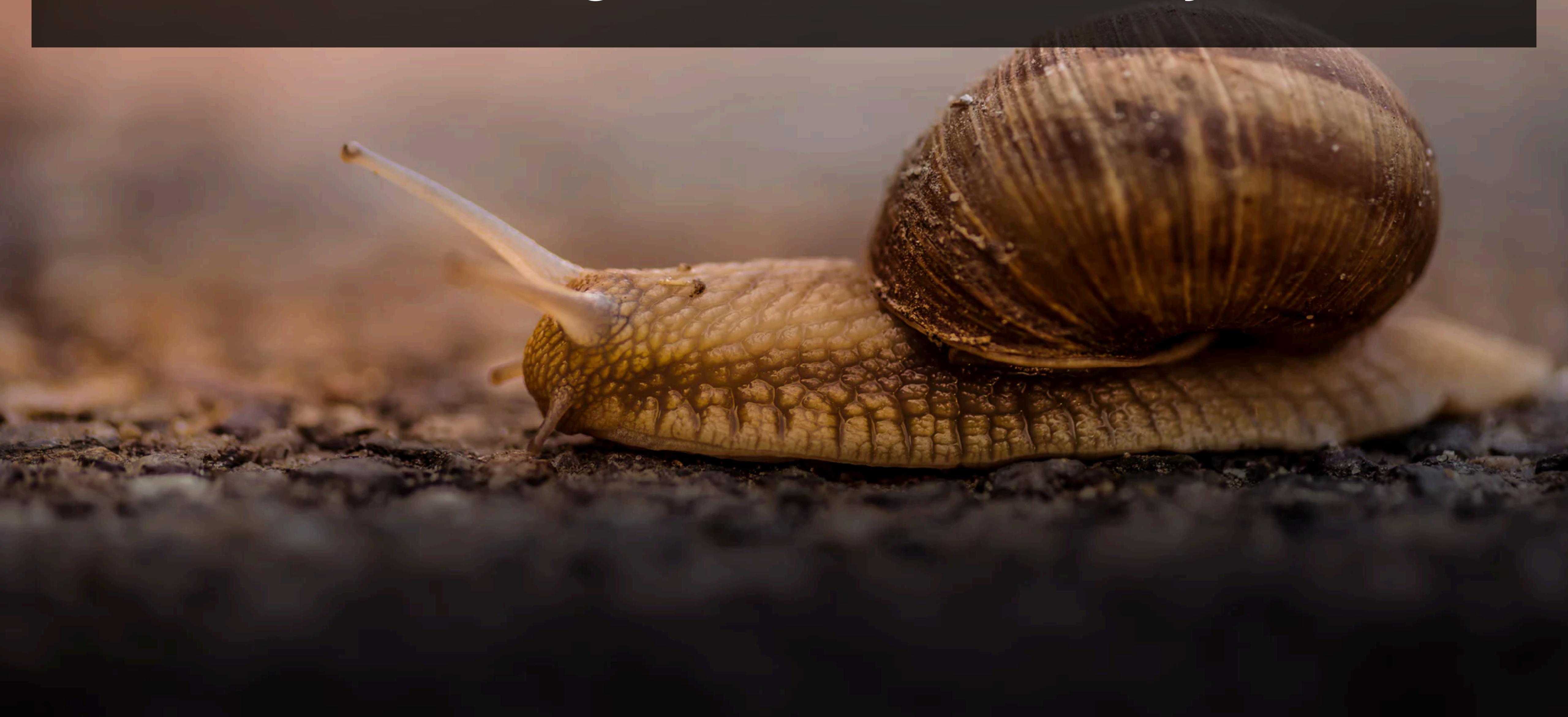








Slowing it down artificially



Application Structure

- An application component
- Two list components
- Slow computation for each entry



Real data...



twitter.com/mgechev

Sales			
Enter name here		Enter name here	
Nettle	196418	trash	Sonja
Rosalie	317811	trash	Analiese
Rhona	514229	trash	Concettina
Talyah	196418	trash	Sherri
Fancie	514229	trash	Emmalee
Kari	317811	trash	Darya
Caresa	514229	trash	Alisun
Gerta	196418	trash	Annamarie
Modestine	196418	trash	Midge

Sales

Enter name here

Nettle
Rosalie
Rhona
Talyah
Fancie
Kari
Caresa
Gerta
Modestine

196418
317811
514229
196418
514229
317811
514229
196418
196418

R&D

Enter name here

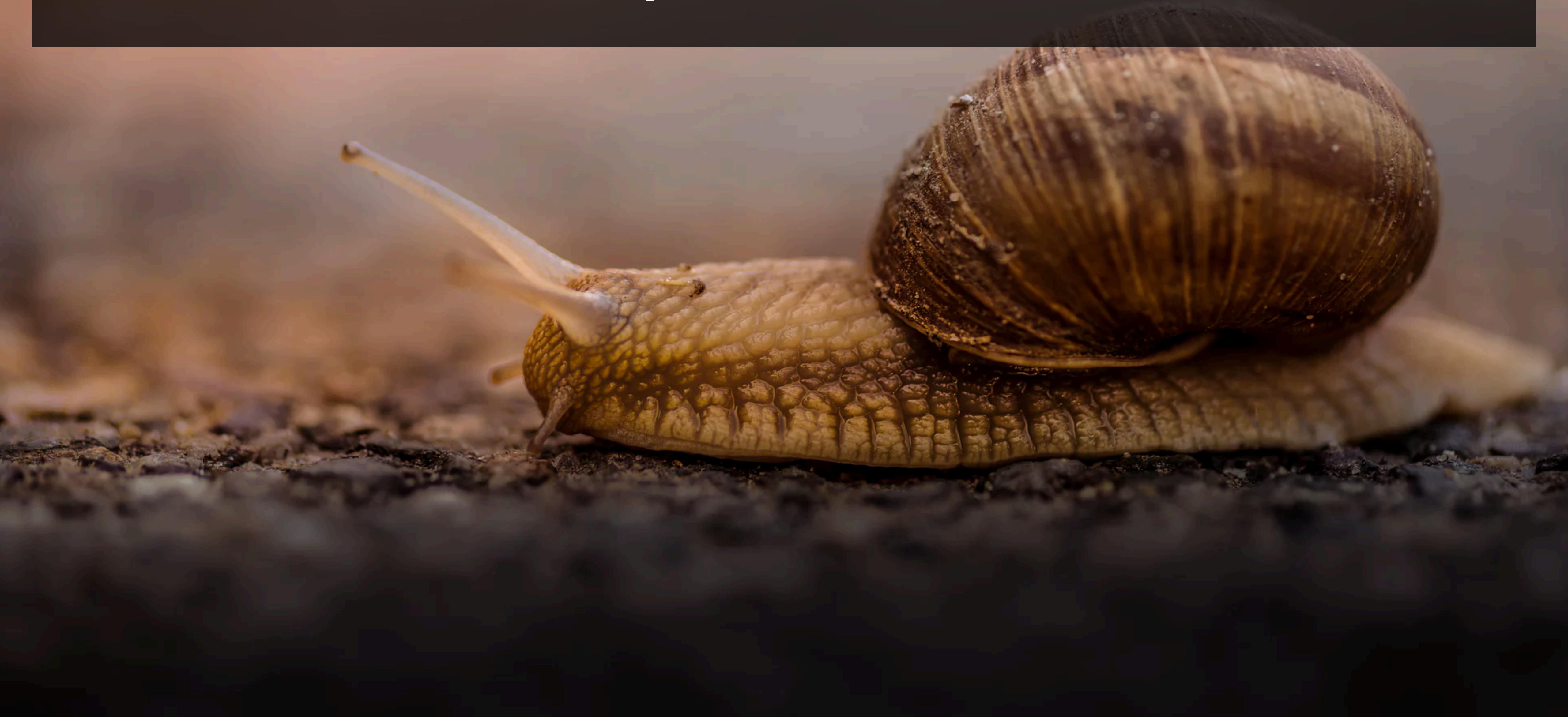
Sonja
Analiese
Concettina
Sherri
Emmalee
Darya
Alisun
Annamarie
Midge

317811
514229
196418
196418
514229
196418
317811
514229
514229

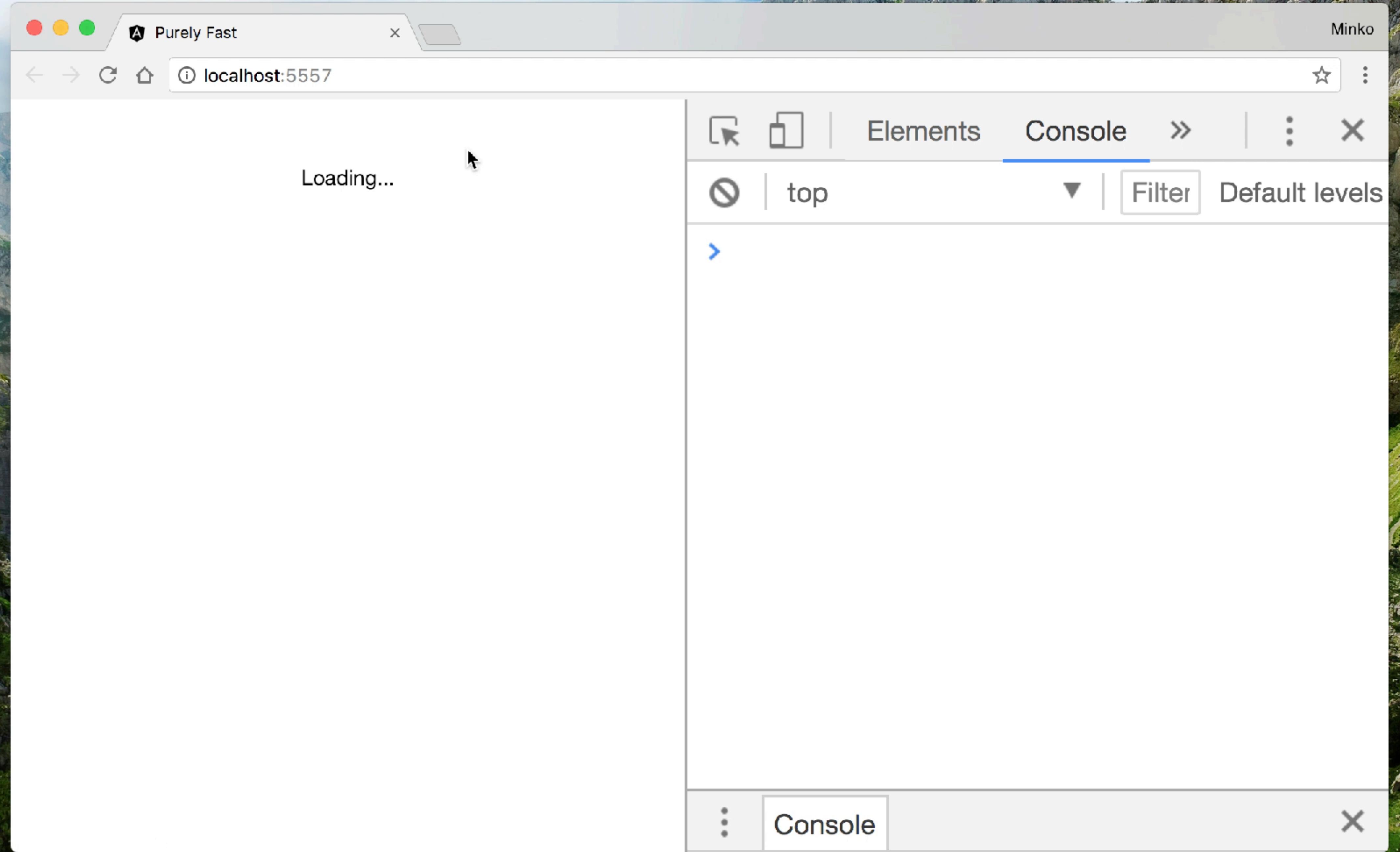
140 entries

Sales			
Enter name here		Enter name here	
Nettle	196418		Sonja
Rosalie	317811		Analiese
Rhona	514229		Concettina
Talyah	196418		Sherri
Fancie	514229		Emmalee
Kari	317811		Darya
Caresa	514229		Alisun
Gerta	196418		Annamarie
Modestine	196418		Midge

Why that slow?



```
@Component(...)  
export class EmployeeListComponent {  
    ...  
    calculate(num: number) {  
        console.log('Computing for entry in',  
            this.department);  
        return fibonacci(num);  
    }  
}
```



Ideas for optimization?



twitter.com/mgechev

OnPush

Change Detection Strategy



twitter.com/mgechev

With OnPush change detection will be triggered when the framework, with reference check, determines that any of the inputs of a component has changed...



What does this mean?

Lets think of EmployeeListComponent as a function, where:

- Inputs are function's arguments
- Rendered component is function's result



Pseudo code (not Angular)

```
const f = EmployeeListComponent;
const data = [e1];

// Will trigger CD
f({ data: data });

data.push(e2);
// Will not trigger CD
f({ data: data });

// Will trigger CD
f({ data: data.slice() });
```

Pseudo code (not Angular)

```
const f = EmployeeListComponent;
const data = [e1];

// Will trigger CD
f({ data: data });

data.push(e2);
// Will not trigger CD
f({ data: data });

// Will trigger CD
f({ data: data.slice() });
```

Pseudo code (not Angular)

```
const f = EmployeeListComponent;
const data = [e1];

// Will trigger CD
f({ data: data });

data.push(e2);
// Will not trigger CD
f({ data: data });

// Will trigger CD
f({ data: data.slice() });
```

Pseudo code (not Angular)

```
const f = EmployeeListComponent;
const data = [e1];

// Will trigger CD
f({ data: data });

data.push(e2);
// Will not trigger CD
f({ data: data });

// Will trigger CD
f({ data: data.slice() });
```

Passing new reference triggers the change detection



twitter.com/mgechev

Should we copy the array every time?



Why would we do that...?



IMMUTABLE



Star

21205

Immutable.js helps:

- We get a new reference on change
- We do not copy the entire data structure



twitter.com/mgechev

```
@Component({
  template: `
    <sd-employee-list
      [data]="list"
      (add)="list = add(list, $event)"
      (remove)="list = remove(list, $event)"
    ></sd-employee-list>
  `
})
export class AppComponent implements OnInit {
  list: List<EmployeeData>;
  add(list: List<EmployeeData>, name: string) {
    return list.unshift({ label: name, num: ... });
  }
  remove(list: List<EmployeeData>, node: EmployeeData) {
    return list.splice(list.indexOf(node), 1);
  }
}
```

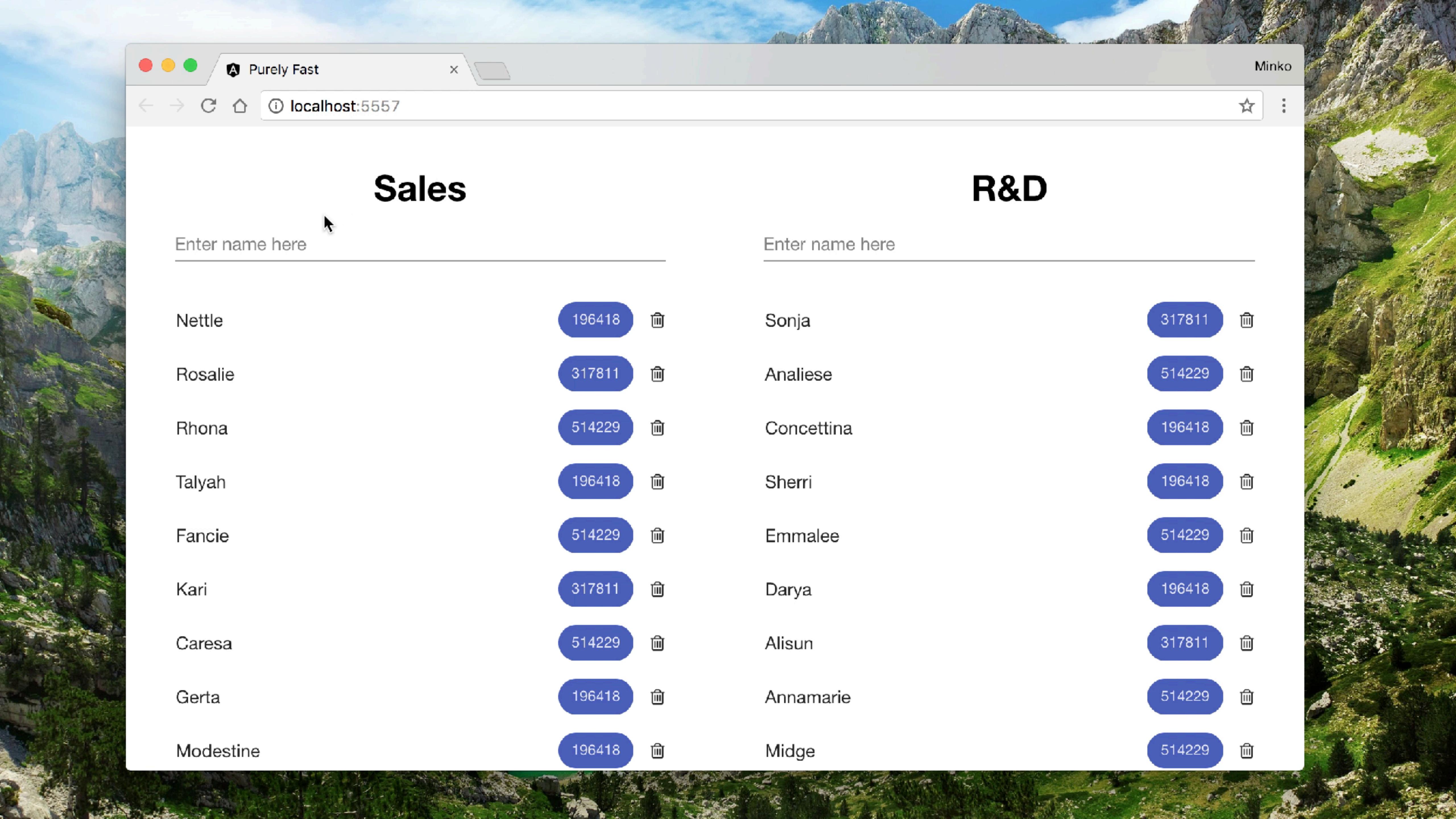
```
@Component({
  template: `
    <sd-employee-list
      [data]="list"
      (add)="list = add(list, $event)"
      (remove)="list = remove(list, $event)"
    ></sd-employee-list>
  `
})
export class AppComponent implements OnInit {
  list: List<EmployeeData>;
  add(list: List<EmployeeData>, name: string) {
    return list.unshift({ label: name, num: ... });
  }
  remove(list: List<EmployeeData>, node: EmployeeData) {
    return list.splice(list.indexOf(node), 1);
  }
}
```

```
@Component({
  template: `
    <sd-employee-list
      [data]="list"
      (add)="list = add(list, $event)"
      (remove)="list = remove(list, $event)"
    ></sd-employee-list>
  `
})
export class AppComponent implements OnInit {
  list: List<EmployeeData>;
  add(list: List<EmployeeData>, name: string) {
    return list.unshift({ label: name, num: ... });
  }
  remove(list: List<EmployeeData>, node: EmployeeData) {
    return list.splice(list.indexOf(node), 1);
  }
}
```

Lets see how fast
it is now!



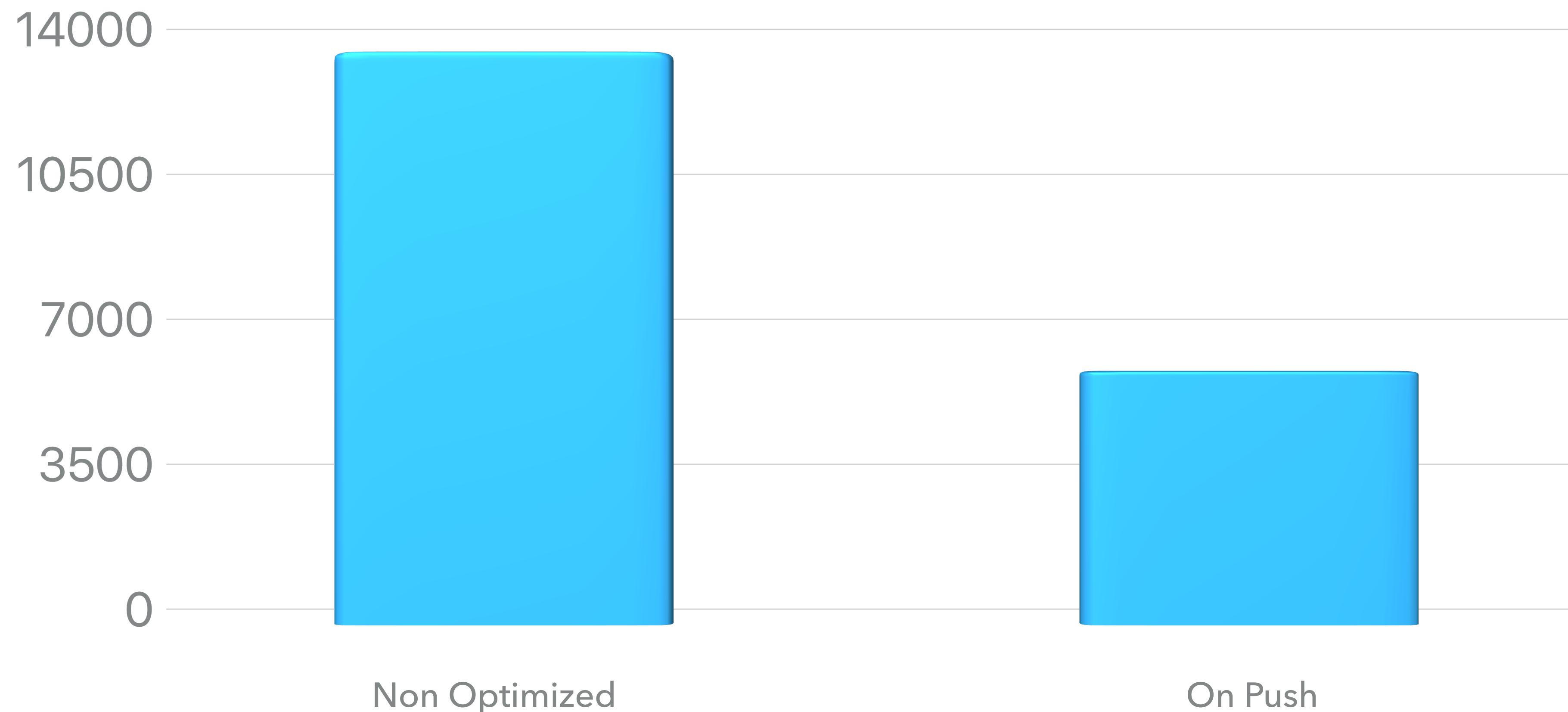
twitter.com/mgechev



A screenshot of a web application titled "Purely Fast" running on "localhost:5557". The application interface is divided into two main sections: "Sales" on the left and "R&D" on the right. Both sections feature a search bar labeled "Enter name here" and a list of names with associated numbers and delete icons.

Sales			
Enter name here			
Nettle	196418	trash	Sonja
Rosalie	317811	trash	Analiese
Rhona	514229	trash	Concettina
Talyah	196418	trash	Sherri
Fancie	514229	trash	Emmalee
Kari	317811	trash	Darya
Caresa	514229	trash	Alisun
Gerta	196418	trash	Annamarie
Modestine	196418	trash	Midge
R&D			
Enter name here			
	317811	trash	
	514229	trash	
	196418	trash	
	196418	trash	
	514229	trash	
	196418	trash	
	317811	trash	
	514229	trash	
	514229	trash	

Typing Speed



2x faster but still slow...



A screenshot of a web browser window titled "Purely Fast" showing a "Sales" page. The page lists names and IDs in blue rounded rectangles with delete icons. A developer tools console is open on the right, showing log entries for each name and ID.

The browser title bar shows "Purely Fast" and the URL "localhost:5557". The developer tools sidebar on the right has tabs for "Elements", "Console", and "Default levels", with "Console" selected. The console log shows entries for each name and ID listed on the page, preceded by a blue arrow icon.

Name	ID
Nettle	196418
Rosalie	317811
Rhona	514229
Talyah	196418
Fancie	514229
Kari	317811
Caresa	514229
Gerta	196418
Modestine	196418

Console Log:

```
> Nettle 196418  
> Rosalie 317811  
> Rhona 514229  
> Talyah 196418  
> Fancie 514229  
> Kari 317811  
> Caresa 514229  
> Gerta 196418  
> Modestine 196418
```

With OnPush change detection will be triggered when the framework, with reference check, determines that any of the inputs of a component has changed...**or when an event in the component is triggered**



Lets do some refactoring!



twitter.com/mgechev

The screenshot shows a web application interface with two main sections: Sales and R&D. Each section contains an input field labeled "Enter name here" and a list of names with their corresponding IDs and a delete icon.

Sales Section:

- Enter name here
- Nettle (196418) delete
- Rosalie (317811) delete
- Rhona (514229) delete
- Talyah (196418) delete
- Fancie (514229) delete
- Kari (317811) delete
- Caresa (514229) delete
- Gerta (196418) delete

R&D Section:

- Enter name here
- Sonja (317811) delete
- Analiese (514229) delete
- Concettina (196418) delete
- Sherri (196418) delete
- Emmalee (514229) delete
- Darya (196418) delete
- Alisun (317811) delete
- Annamarie (514229) delete

AppComponent

EmployeesListComponent

NameInputComponent

ListComponent

The screenshot shows a web application interface with two main sections: "Sales" and "R&D".

Sales Section:

- Header: "Sales"
- Search Bar: "Enter name here"
- Data Table:

Name	ID	Action
Nettle	196418	Delete
Rosalie	317811	Delete
Rhona	514229	Delete
Talyah	196418	Delete
Fancie	514229	Delete
Kari	317811	Delete
Caresa	514229	Delete
Gerta	196418	Delete

R&D Section:

- Header: "R&D"
- Search Bar: "Enter name here"
- Data Table:

Name	ID	Action
Sonja	317811	Delete
Analiese	514229	Delete
Concettina	196418	Delete
Sherri	196418	Delete
Emmalee	514229	Delete
Darya	196418	Delete
Alisun	317811	Delete
Annamarie	514229	Delete

AppComponent

EmployeesListComponent

NameInputComponent

ListComponent

A screenshot of a web application titled "Purely Fast" running on "localhost:5557". The application has two main sections: "Sales" and "R&D".

Sales Section:

- Enter name here
- Nettle 196418 Delete
- Rosalie 317811 Delete
- Rhona 514229 Delete
- Talyah 196418 Delete
- Fancie 514229 Delete
- Kari 317811 Delete
- Caresa 514229 Delete
- Gerta 196418 Delete

R&D Section:

- Enter name here
- Sonja 317811 Delete
- Analiese 514229 Delete
- Concettina 196418 Delete
- Sherri 196418 Delete
- Emmalee 514229 Delete
- Darya 196418 Delete
- Alisun 317811 Delete
- Annamarie 514229 Delete

AppComponent

EmployeesListComponent

NameInputComponent

ListComponent

Purely Fast

localhost:5557

Sales

Enter name here

Nettle	196418	trash
Rosalie	317811	trash
Rhona	514229	trash
Talyah	196418	trash
Fancie	514229	trash
Kari	317811	trash
Caresa	514229	trash
Gerta	196418	trash

R&D

Enter name here

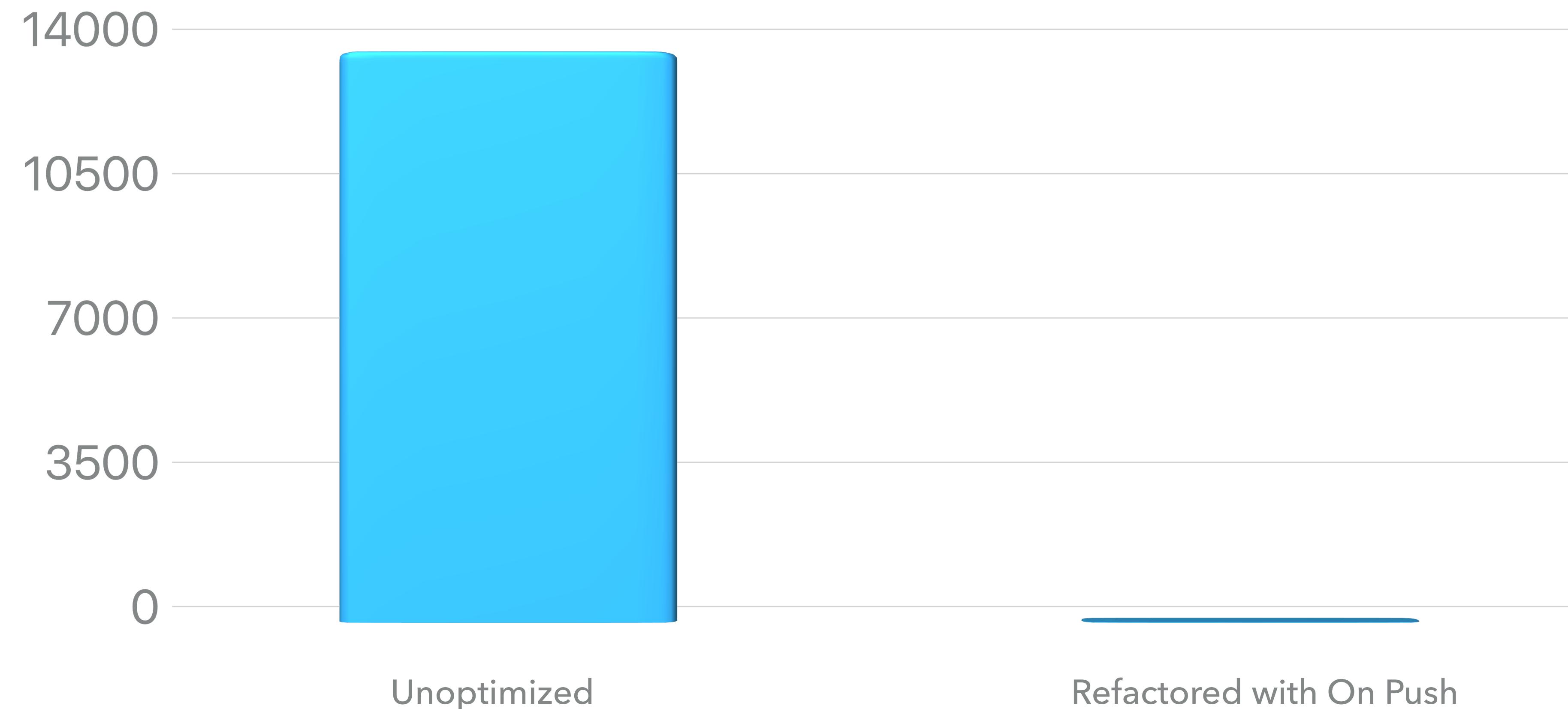
Sonja	317811	trash
Analiese	514229	trash
Concettina	196418	trash
Sherri	196418	trash
Emmalee	514229	trash
Darya	196418	trash
Alisun	317811	trash
Annamarie	514229	trash

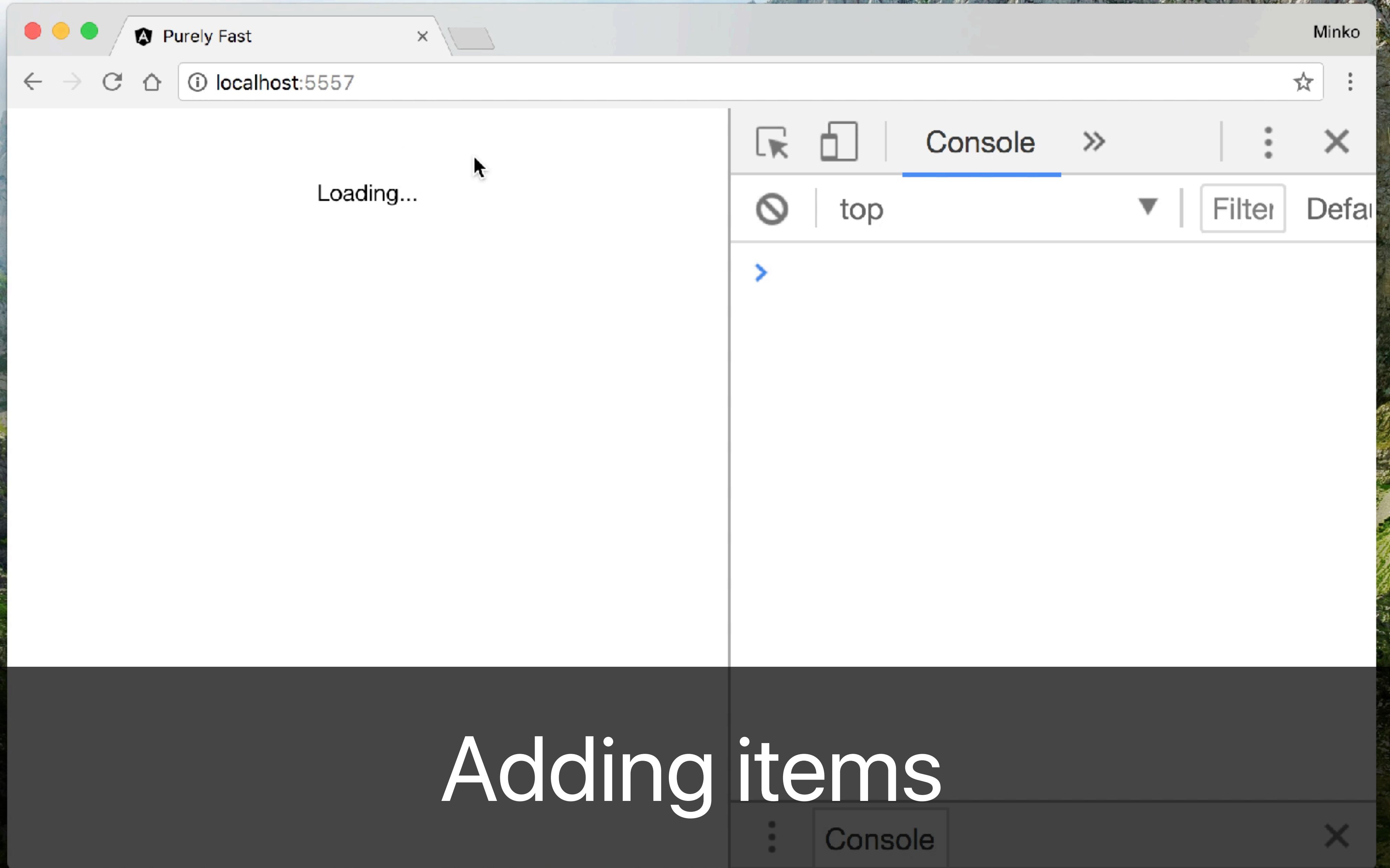
Sooo much faster!



twitter.com/mgechev

Typing Speed





Recomputing everything
every time we add a new entry



```
const fibonacci = n => {
  if (n === 1 || n === 2) return 1;
  return fibonacci(n - 1) + fibonacci(n - 2);
};
```

```
const fibonacci = n => {
  if (n === 1 || n === 2) return 1;
  return fibonacci(n - 1) + fibonacci(n - 2);
};
```

// Two properties

// - No side effects

// - Same result for same arguments

Pure Function



twitter.com/mgechev

Pipes in Angular

- Pure
- Impure



twitter.com/mgechev

Angular executes a pure pipe only when it detects a change to the input value. A pure change is either a change to a primitive input value (String, Number, Boolean, Symbol) or a changed object reference (Date, Array, Function, Object).



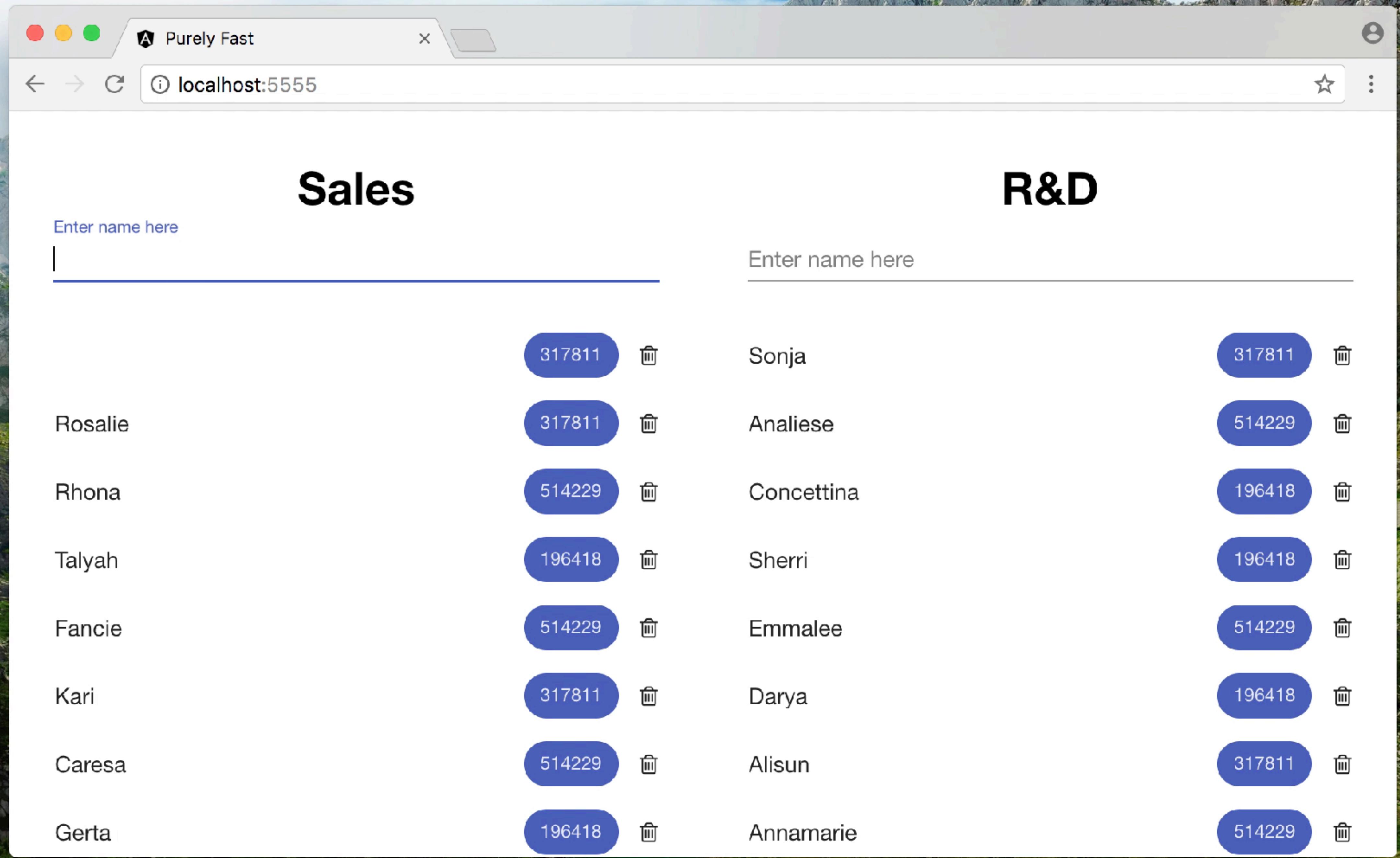
```
@Pipe({ name: 'calculate', pure: true })
export class CalculatePipe {
  transform(num: number) {
    return fibonacci(num);
  }
}
```

```
@Component({
  changeDetection: ChangeDetectionStrategy.OnPush,
  template: `
    ...
    {{ item.num | calculate }}
    ...
  `
})
export classListComponent { ... }
```

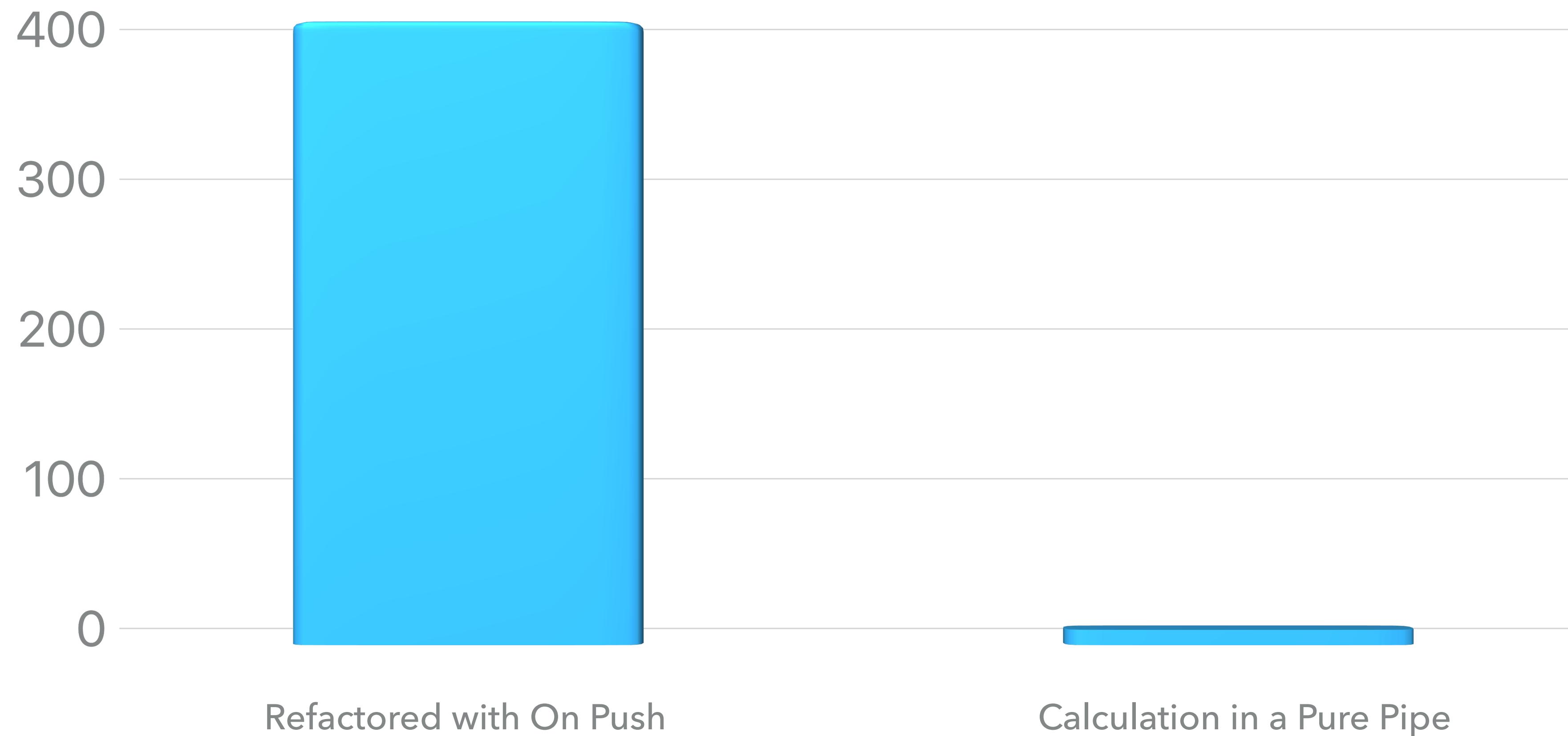
Lets benchpress it!



twitter.com/mgechev



Adding / Removing Entries



Initial rendering...

with 1000 items



Purely Fast

localhost:5557

Elements Network Minko

Loading...

Filter Regex Hide data URLs

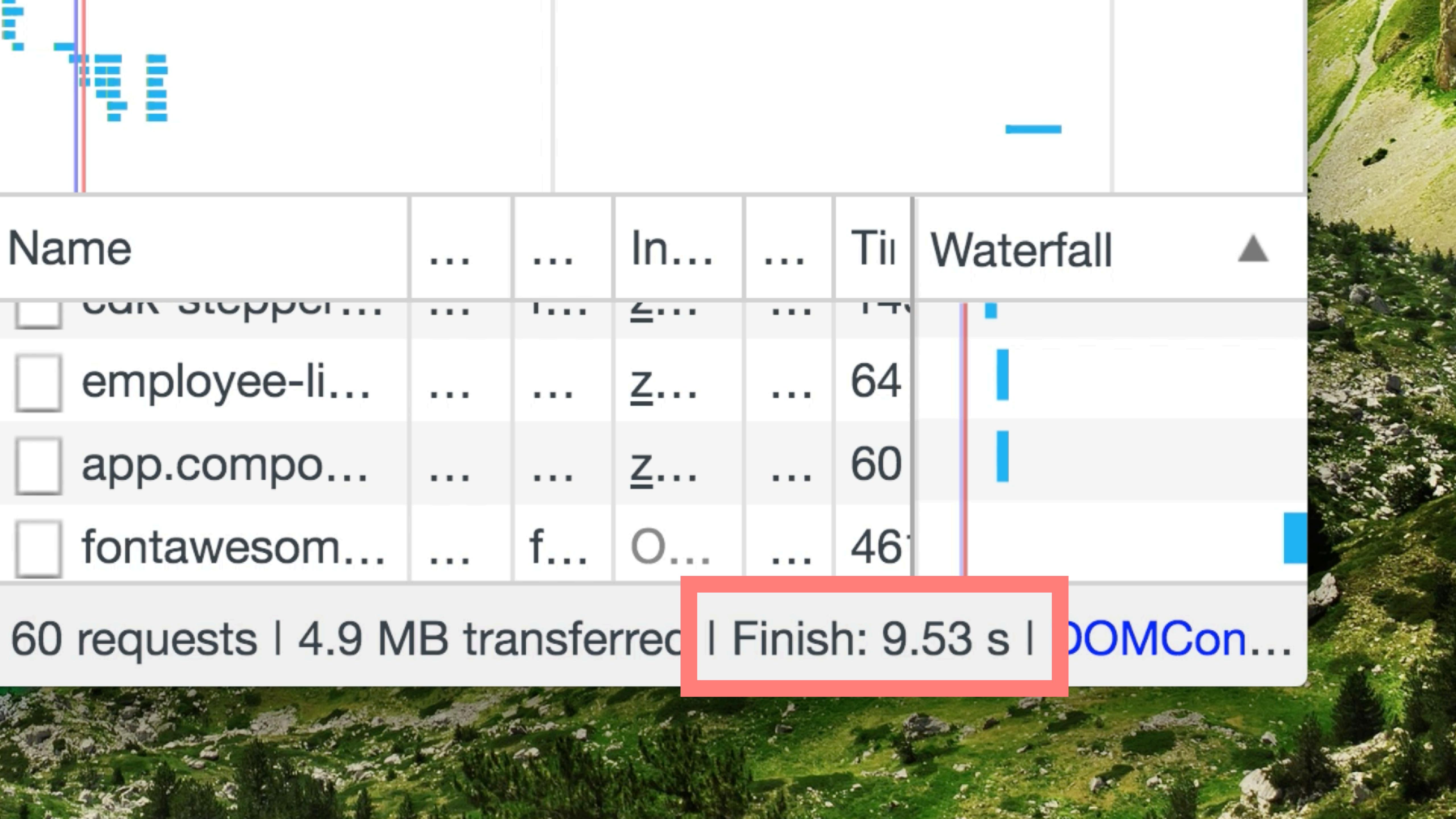
All XHR JS CSS Img Media Font Doc WS Manifest

50 ms 100 ms

Name	In...	...	Til	Waterfall
localhost	O...	...	16	

1 requests | 2.6 KB transferred | Finish: 16 ms

This screenshot shows a browser developer tools Network tab overlying a scenic mountain landscape. The Network tab displays a single request to 'localhost' which completed in 16ms. The tab includes filters for XHR, JS, CSS, Img, Media, Font, Doc, WS, and Manifest resources, with 'All' selected. A watermark for 'Minko' is visible in the top right corner of the developer tools interface.



A screenshot of a performance monitoring application showing network traffic details. The table lists various components and their metrics. A tooltip at the bottom provides summary statistics.

Name	In...	...	Til	Waterfall
car.stopper...	≤...	...	17	
employee-li...	≥...	...	64	
app.compo...	≥...	...	60	
fontawesom...	...	f...	O...	...	46	

60 requests | 4.9 MB transferred | I Finish: 9.53 s | DOMCon...

Lets take a look at our data



twitter.com/mgechev

Sales

Enter name here

Nettle 196418 

Rosalie 317811 

Rhona 514229 

Talyah 196418 

Fancie 514229 

Kari 317811 

Caresa 514229 

Gerta 196418 

Sales

Enter name here

Nettle

196418



Rosalie

317811



Rhona

514229



Talyah

196418



Fancie

514229



Kari

317811



Caresa

514229

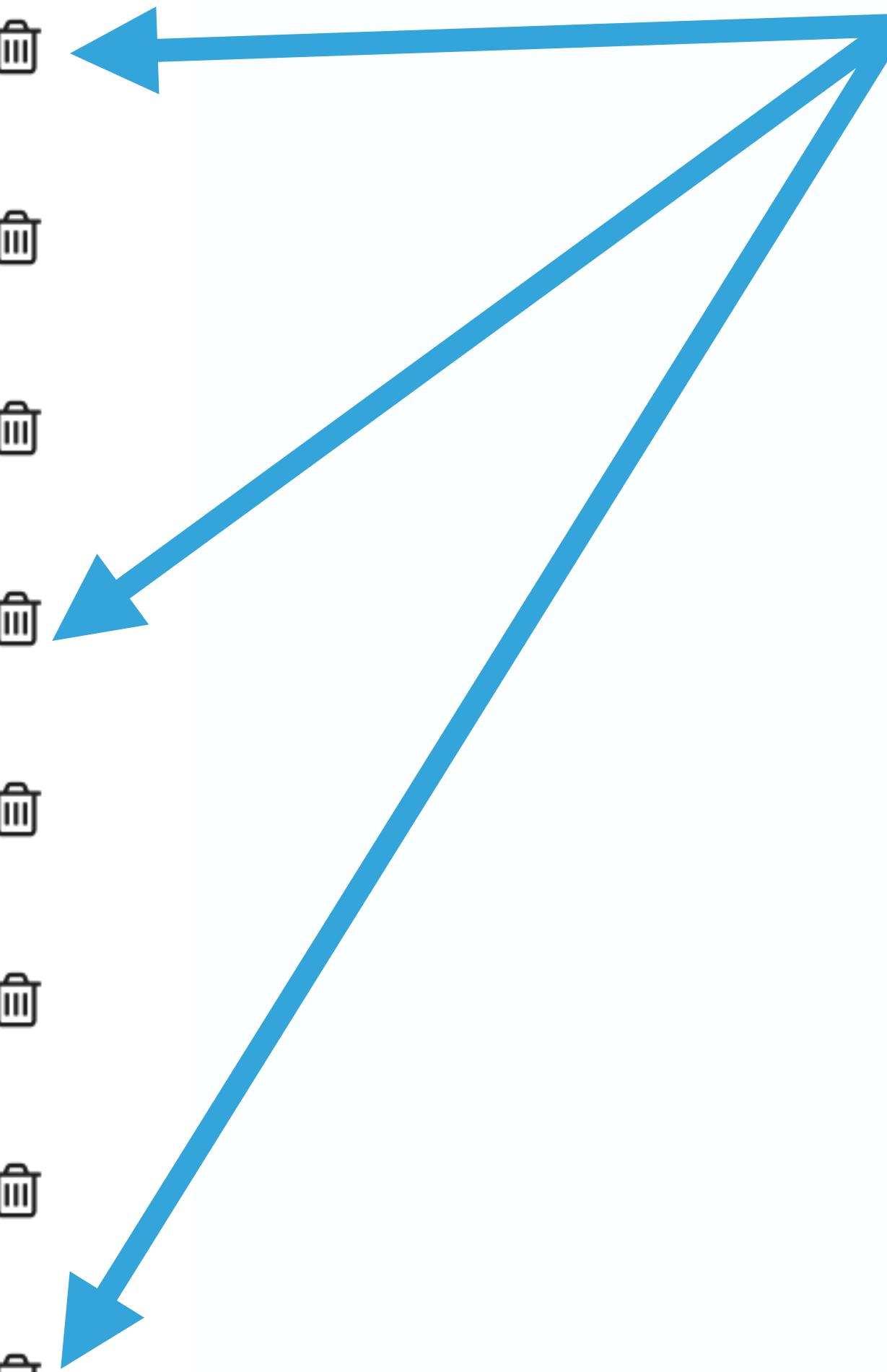


Gerta

196418



fibonacci(27)



Sales

Enter name here

Nettle

196418



Rosalie

317811



Rhona

514229



Talyah

196418



Fancie

514229



Kari

317811



Caresa

514229



Gerta

196418



fibonacci(27)

fibonacci(28)

196418



317811



514229



196418



514229



317811



514229



196418



Sales

Enter name here

Nettle

196418



fibonacci(27)

Rosalie

317811



fibonacci(28)

Rhona

514229



Talyah

196418



Fancie

514229



Kari

317811



Caresa

514229



Gerta

196418



fibonacci(29)

196418



317811



514229



196418



514229



317811



514229



196418



196418



317811



514229



196418



514229



317811



514229



196418



Samples from a small range

During initial rendering we recompute
same value multiple times



Solution

Caching the value once computed



twitter.com/mgechev

Memoization



twitter.com/mgechev

Memoization possible for pure functions



twitter.com/mgechev

```
const memoize = require('lodash.memoize');

const fibonacci = memoize((num: number): number => {
  if (num === 1 || num === 2) return 1;
  return fibonacci(num - 1) + fibonacci(num - 2);
});
```

```
const memoize = require('lodash.memoize');

const fibonacci = memoize((num: number): number => {
  if (num === 1 || num === 2) return 1;
  return fibonacci(num - 1) + fibonacci(num - 2);
});
```

Purely Fast

localhost:5557

Minko

Elements Network

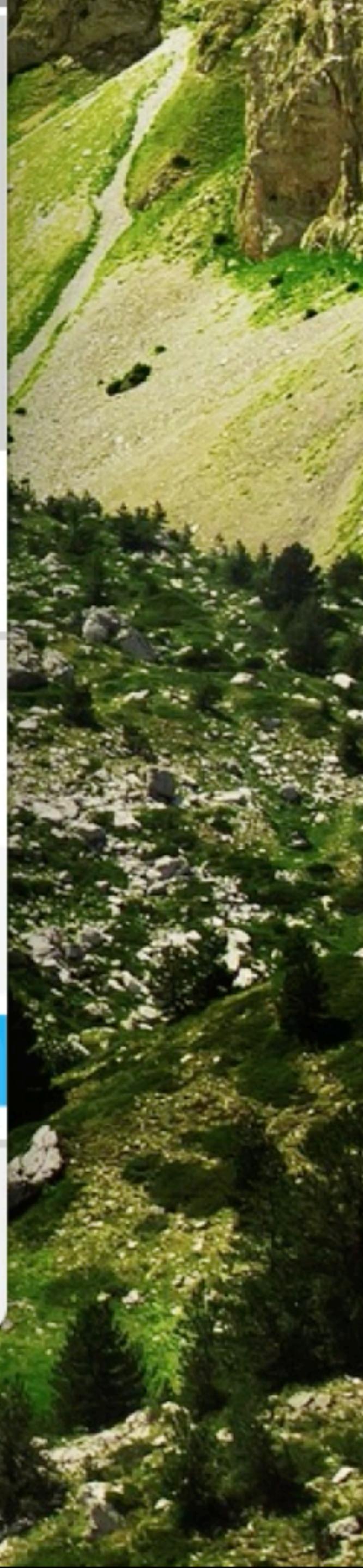
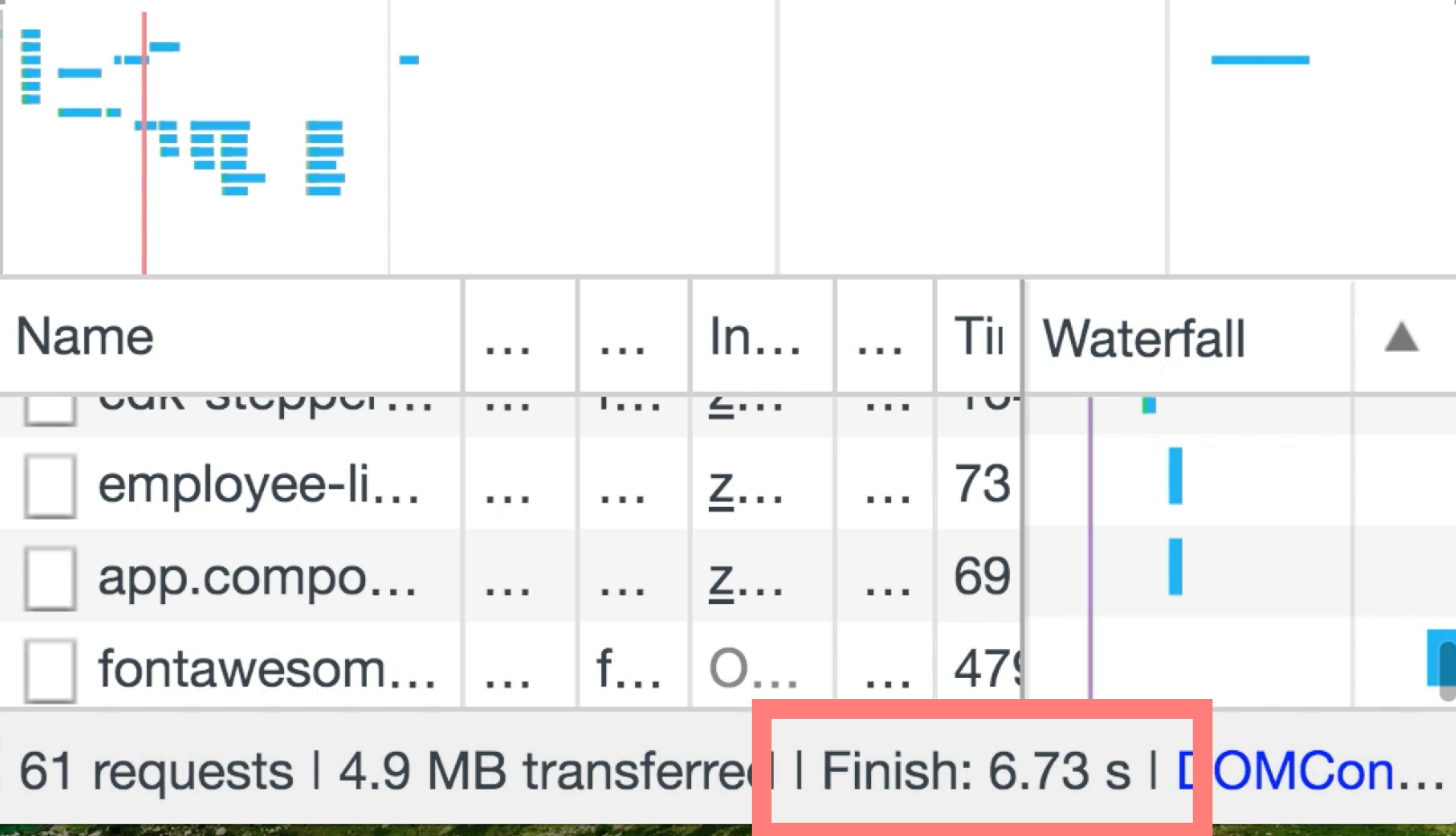
Filter Regex Hide data URLs

All XHR JS CSS Img Media Font Doc WS Manifest

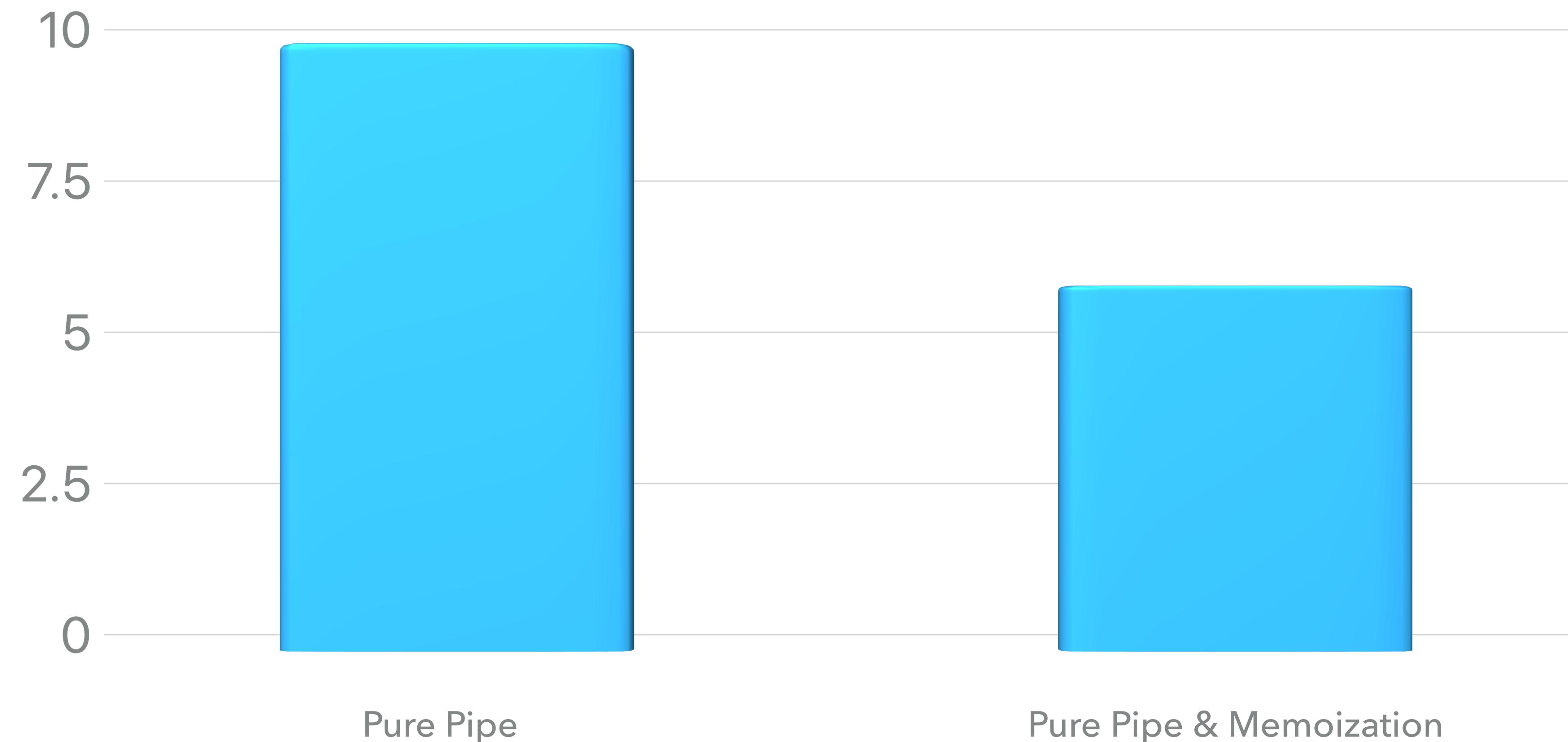
100 ms 200 ms

Name	In...	...	Til	Waterfall
Intl.min.js?1...	↓...	...	15	
system.src.j...	↓...	...	72	
system-con...	↓...	...	72	

11 requests | 231 KB transferred | Finish: 210 ms



Initial Rendering



Pure Pipes

Nettle

196418



27 | calculate

Rosalie

317811



Rhona

514229



Talyah

196418



27 | calculate

Fancie

514229



Kari

317811



Caresa

514229



Gerta

196418



27 | calculate

Pure Pipes

Nettle	196418	trash	27 calculate → fib(27) → 196418
Rosalie	317811	trash	
Rhona	514229	trash	
Talyah	196418	trash	27 calculate
Fancie	514229	trash	
Kari	317811	trash	
Caresa	514229	trash	
Gerta	196418	trash	27 calculate

Pure Pipes

Nettle	196418	trash	27 calculate → fib(27) → 196418
Rosalie	317811	trash	
Rhona	514229	trash	
Talyah	196418	trash	27 calculate → fib(27) → 196418
Fancie	514229	trash	
Kari	317811	trash	
Caresa	514229	trash	
Gerta	196418	trash	27 calculate

Pure Pipes

Nettle	196418	trash	27 calculate → fib(27) → 196418
Rosalie	317811	trash	
Rhona	514229	trash	
Talyah	196418	trash	27 calculate → fib(27) → 196418
Fancie	514229	trash	
Kari	317811	trash	
Caresa	514229	trash	
Gerta	196418	trash	27 calculate → fib(27) → 196418

Memoization

Nettle

196418



27 | calculate

Rosalie

317811



Rhona

514229



Talyah

196418



27 | calculate

Fancie

514229



Kari

317811



Caresa

514229



Gerta

196418



27 | calculate

Memoization

Nettle

196418



27 | calculate → fib(27) → 196418

Rosalie

317811



Rhona

514229



Talyah

196418



27 | calculate

Fancie

514229



Kari

317811



Caresa

514229



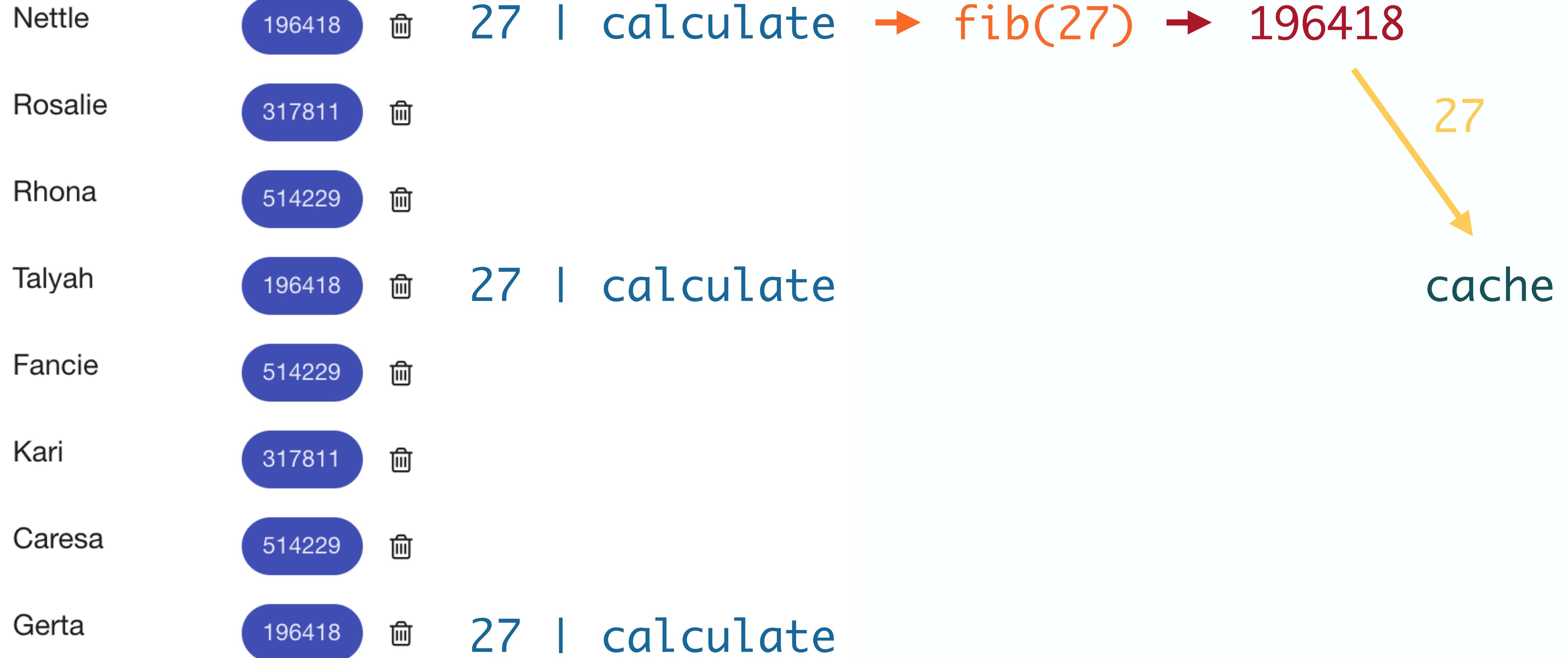
Gerta

196418

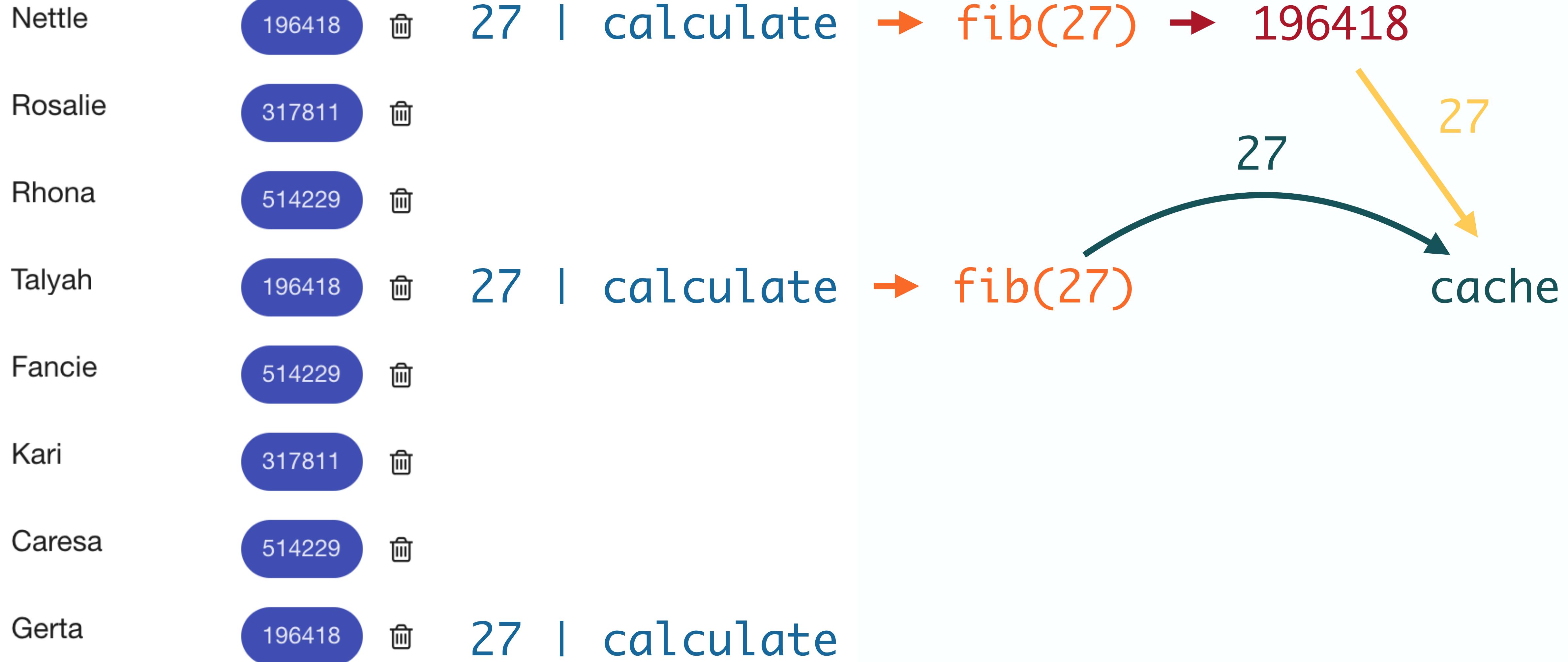


27 | calculate

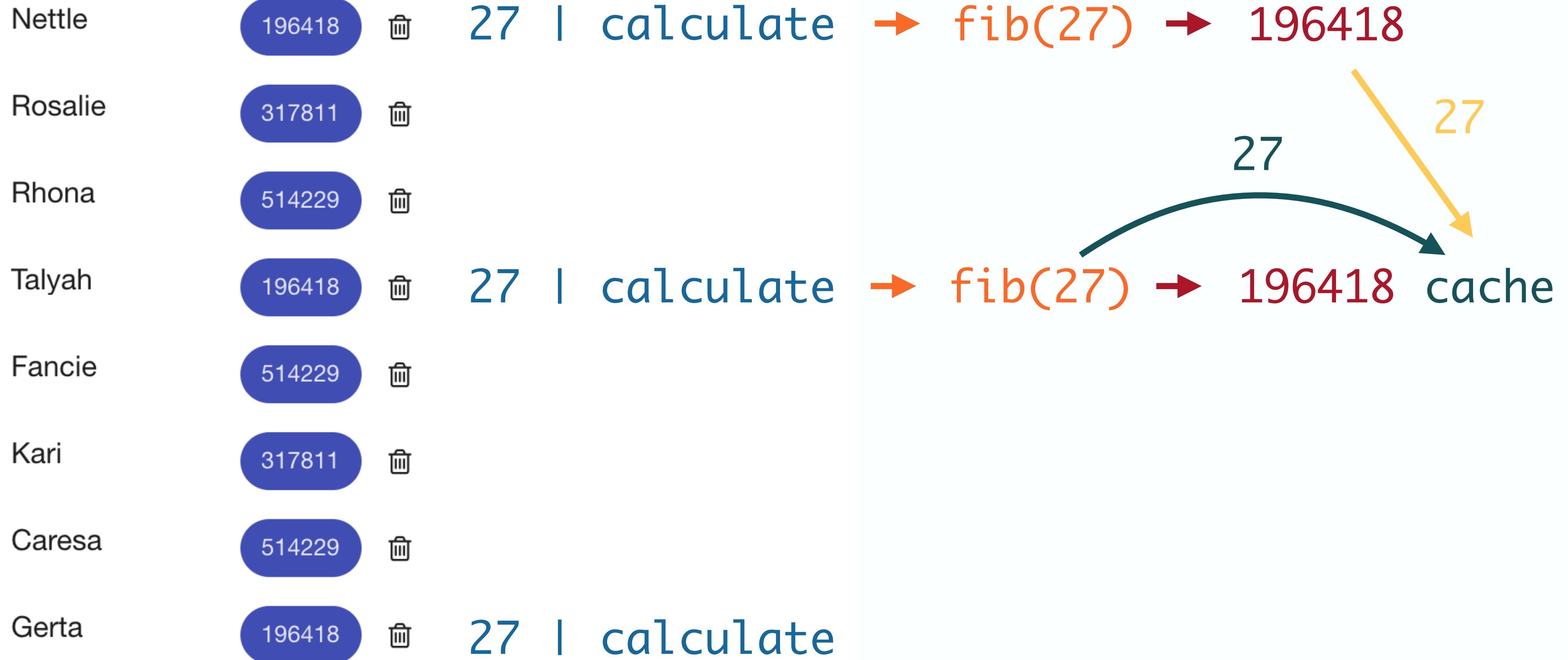
Memoization



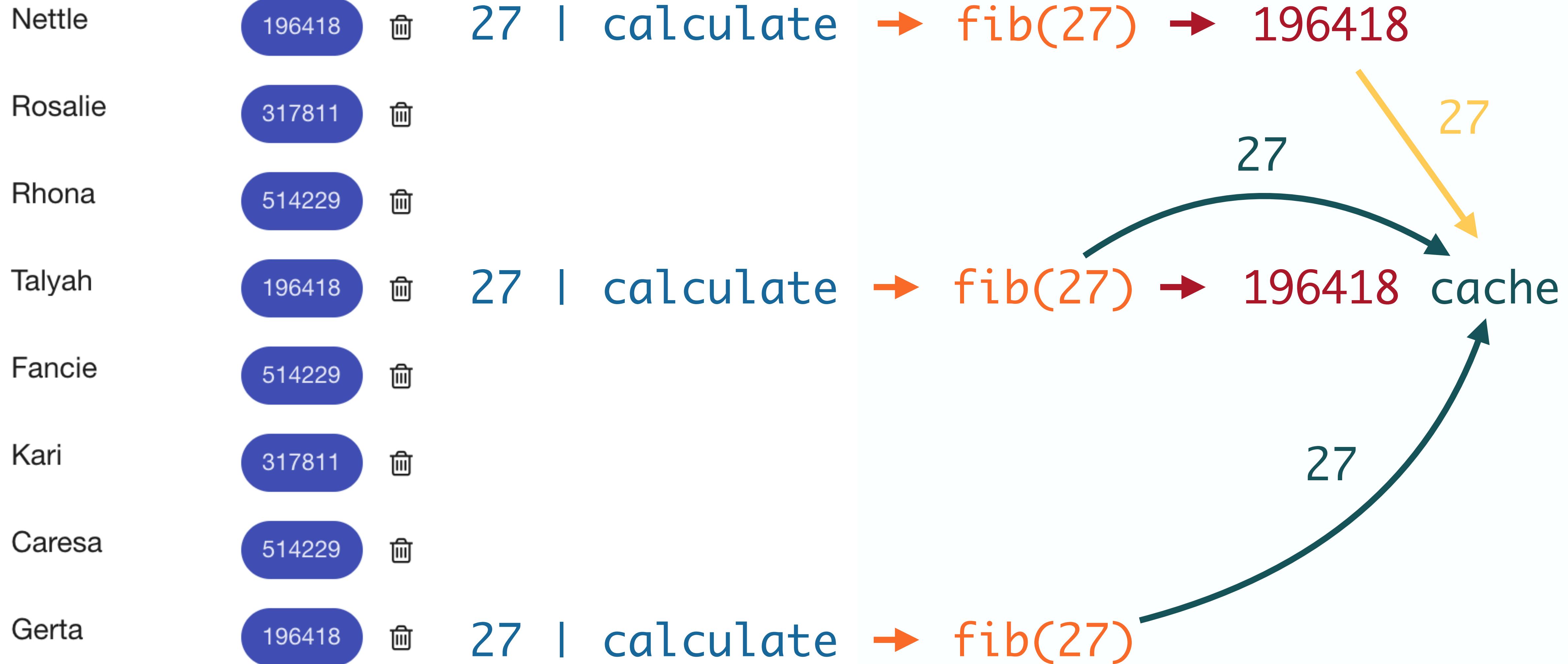
Memoization



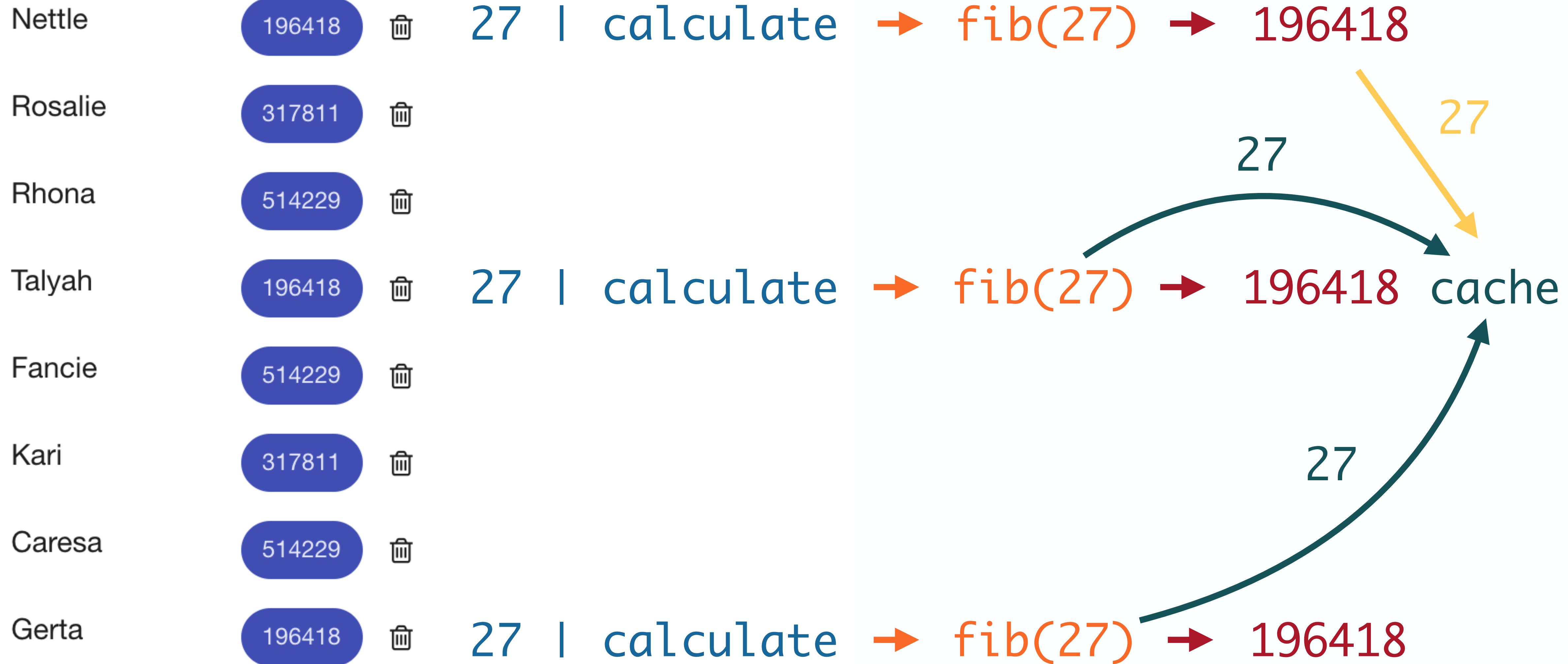
Memoization



Memoization



Memoization



Pattern...

- On push performs “memoization”
- Pure pipes are “memoized”



twitter.com/mgechev

Pattern...

- On push performs “memoization”
- Pure pipes are “memoized”

for their last input



Lets try to do better!



twitter.com/mgechev

How NgForOf works

```
@Directive({selector: '[ngFor][ngForOf]'})
export class NgForOf<T> implements DoCheck, OnChanges {
  ...
  constructor(private _differs: IterableDiffer) {}

  ngDoCheck(): void {
    const changes = this._differ.diff(this.ngFor);
    if (changes) this._applyChanges(changes);
  }
  ...
}
```

How NgForOf works

```
@Directive({selector: '[ngFor][ngForOf]'})
export class NgForOf<T> implements DoCheck, OnChanges {
  ...
  constructor(private _differs: IterableDiffer) {}

  ngDoCheck(): void {
    const changes = this._differ.diff(this.ngFor);
    if (changes) this._applyChanges(changes);
  }
  ...
}
```

How NgForOf works

```
@Directive({selector: '[ngFor][ngForOf]'})
export class NgForOf<T> implements DoCheck, OnChanges {
  ...
  constructor(private _differs: IterableDiffer) {}

  ngDoCheck(): void {
    const changes = this._differ.diff(this.ngFor);
    if (changes) this._applyChanges(changes);
  }
  ...
}
```

IterableDiffer checks whether the data structure has changed



But the data structure
knows that best!



```
export class DifferableList<T> {
  changes = new LinkedList<IterableChangeRecord<T>>();

  constructor(private data = List<T>([])) {}

  unshift(data: T) {
    const result = new DifferableList<T>(this.data.unshift(data));
    result.changes.add({ ... });
    return result;
  }

  ...

  [Symbol.iterator]() {
    return new DifferableListIterator<T>(this);
  }
}
```

```
export class DifferableList<T> {
  changes = new LinkedList<IterableChangeRecord<T>>();

  constructor(private data = List<T>([])) {}

  unshift(data: T) {
    const result = new DifferableList<T>(this.data.unshift(data));
    result.changes.add({ ... });
    return result;
  }

  ...

  [Symbol.iterator]() {
    return new DifferableListIterator<T>(this);
  }
}
```

```
export class DifferableList<T> {
  changes = new LinkedList<IterableChangeRecord<T>>();

  constructor(private data = List<T>([])) {}

  unshift(data: T) {
    const result = new DifferableList<T>(this.data.unshift(data));
    result.changes.add({ ... });
    return result;
  }

  ...

  [Symbol.iterator]() {
    return new DifferableListIterator<T>(this);
  }
}
```

```
export class DifferableList<T> {
  changes = new LinkedList<IterableChangeRecord<T>>();

  constructor(private data = List<T>([])) {}

  unshift(data: T) {
    const result = new DifferableList<T>(this.data.unshift(data));
    result.changes.add({ ... });
    return result;
  }

  ...

  [Symbol.iterator]() {
    return new DifferableListIterator<T>(this);
  }
}
```

```
export class DifferableList<T> {  
  changes = new LinkedList<IterableChangeRecord<T>>();  
  
  constructor(private data = List<T>([])) {}  
  
  unshift(data: T) {  
    const result = new DifferableList<T>(this.data.unshift(data));  
    result.changes.add({ ... });  
    return result;  
  }  
  
  ...  
  
  [Symbol.iterator]() {  
    return new DifferableListIterator<T>(this);  
  }  
}
```

Data structure optimized for Angular



twitter.com/mgechev

```
export class DifferableListDiffer<V>
  implements IterableDiffer<V>, IterableChanges<V> {
  ...
  diff(collection: NgIterable<V>): DifferableListDiffer<V> | null {
    const changes = this._data.changes;
    this._changes = changes;
    if (changes.size() > 0) {
      this._data.changes = new LinkedList<IterableChangeRecord<V>>();
      return this;
    } else {
      return null;
    }
  }
}
```

```
export class DifferableListDiffer<V>
  implements IterableDiffer<V>, IterableChanges<V> {
  ...
  diff(collection: NgIterable<V>): DifferableListDiffer<V> | null {
    const changes = this._data.changes;
    this._changes = changes;
    if (changes.size() > 0) {
      this._data.changes = new LinkedList<IterableChangeRecord<V>>();
      return this;
    } else {
      return null;
    }
  }
}
```

Required refactoring



twitter.com/mgechev

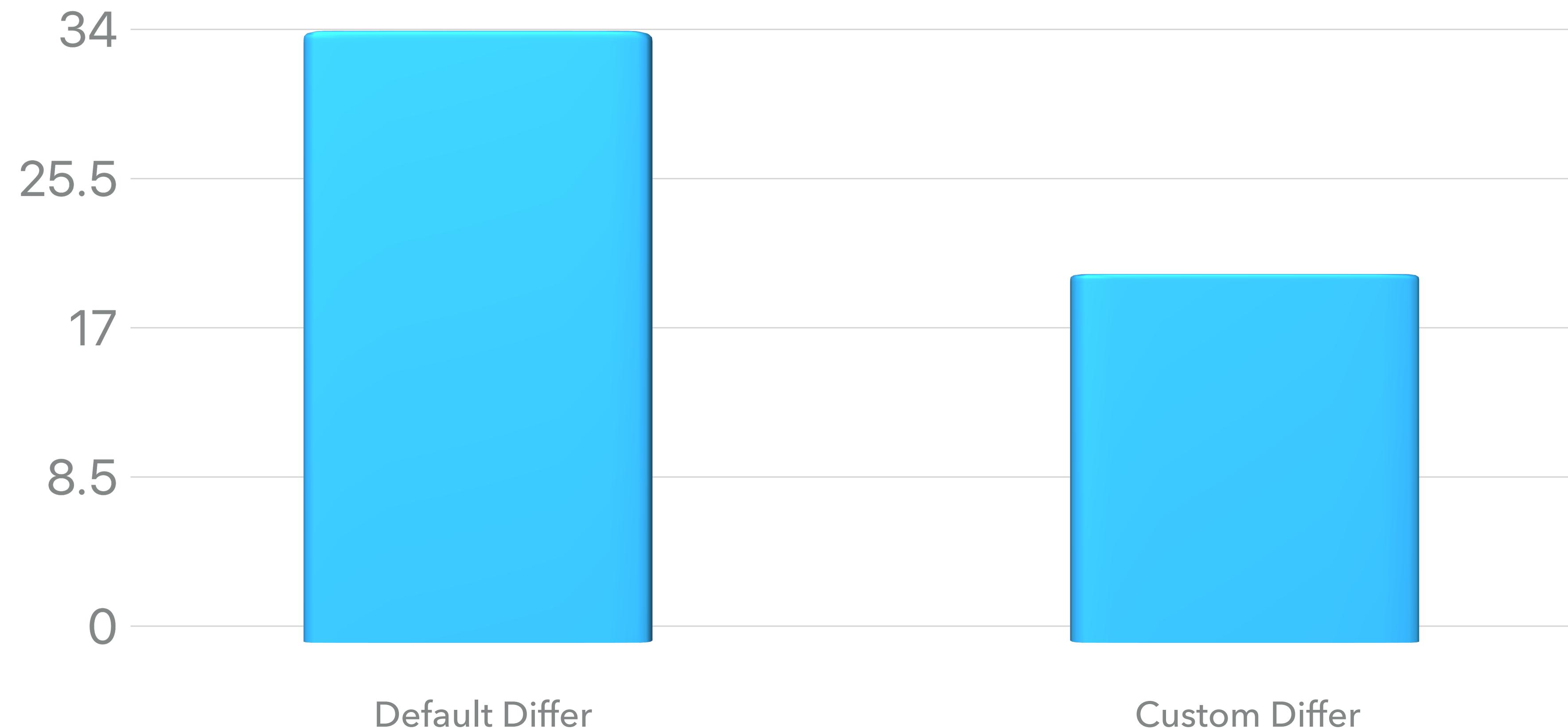
```
@Component({
  selector: 'sd-employee-list',
  providers: [
    IterableDifferers.extend([
      new DifferableListDifferFactory()
    ])
  ],
  template: `...`,
})
export class EmployeeListComponent { ... }
```

Inspired by
Persistent
data structures



twitter.com/mgechev

Adding / Removing Entries



Lessons learned

- No silver bullet
- Understand your component structure
- Understand your data
- Application specific benchmarks



twitter.com/mgechev



Get inspiration from computer science



twitter.com/mgechev

Links

mgv.io/ng-cd – Angular's OnPush Change Detection Strategy

mgv.io/ng-pure – Pure Pipes and Referential Transparency

mgv.io/ng-diff – Understanding Angular Differs

mgv.io/ng-perf-checklist – Angular Performance Checklist



Thank you!



- [!\[\]\(613283d7582b9d5247dedbb7d8b54f8b_img.jpg\) twitter.com/mgechev](https://twitter.com/mgechev)
- [!\[\]\(bf6dc42ef3a9802d0ad3085aedf38663_img.jpg\) github.com/mgechev](https://github.com/mgechev)
- [!\[\]\(1e7e8a1a4d40d09409c9cb236e5c3208_img.jpg\) blog.mgechev.com](https://blog.mgechev.com)