

Software Engineering

Paulo Borba
Informatics Center
Federal University of Pernambuco

pauloborba.cin.ufpe.br

To do before class

- Watch videos
- Read chapter 7 and basic concepts of chapter 6 in the textbook
- Send questions and opinions through slack

SaaS I

Software Engineering

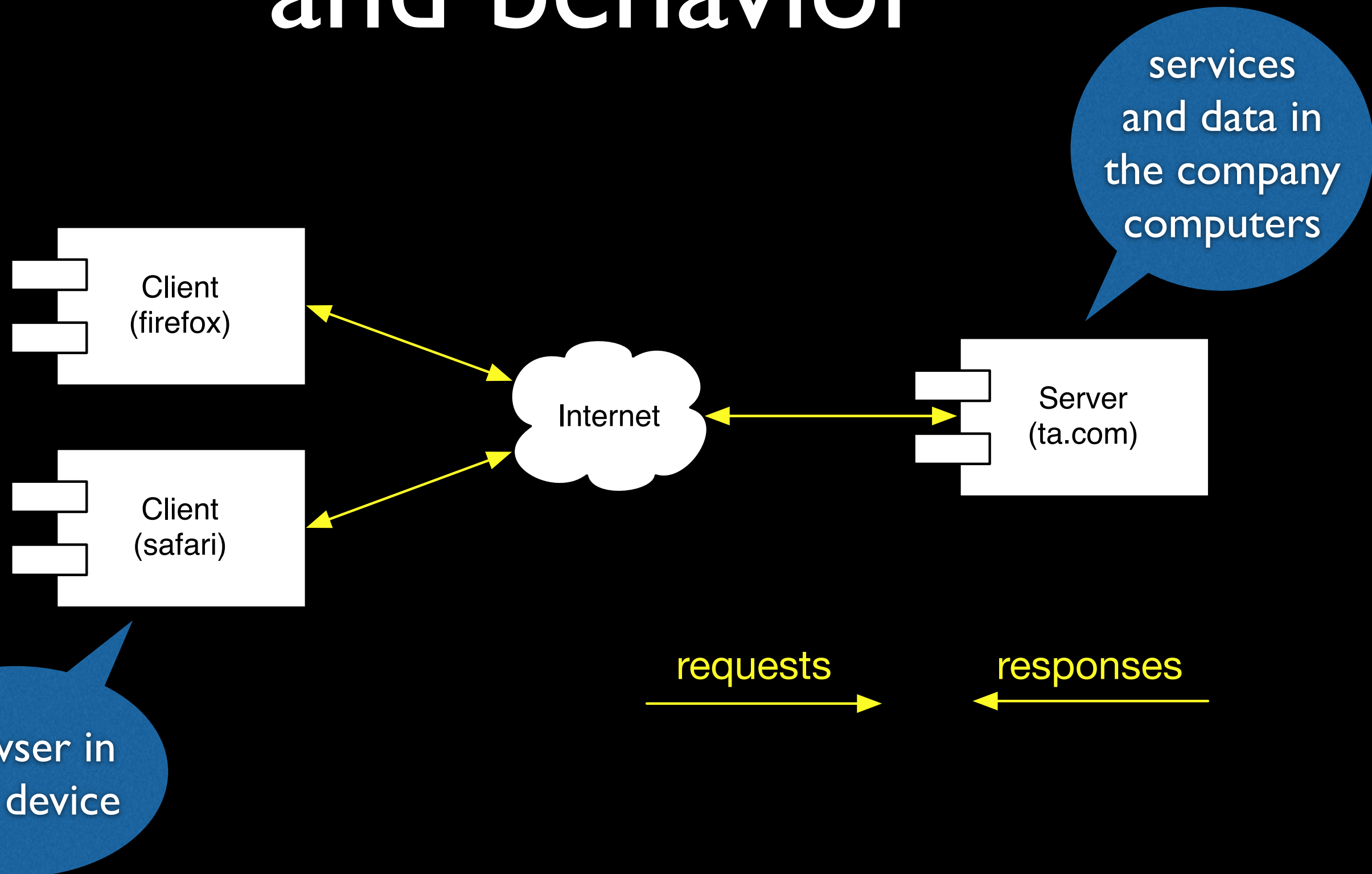
Paulo Borba
Informatics Center
Federal University of Pernambuco

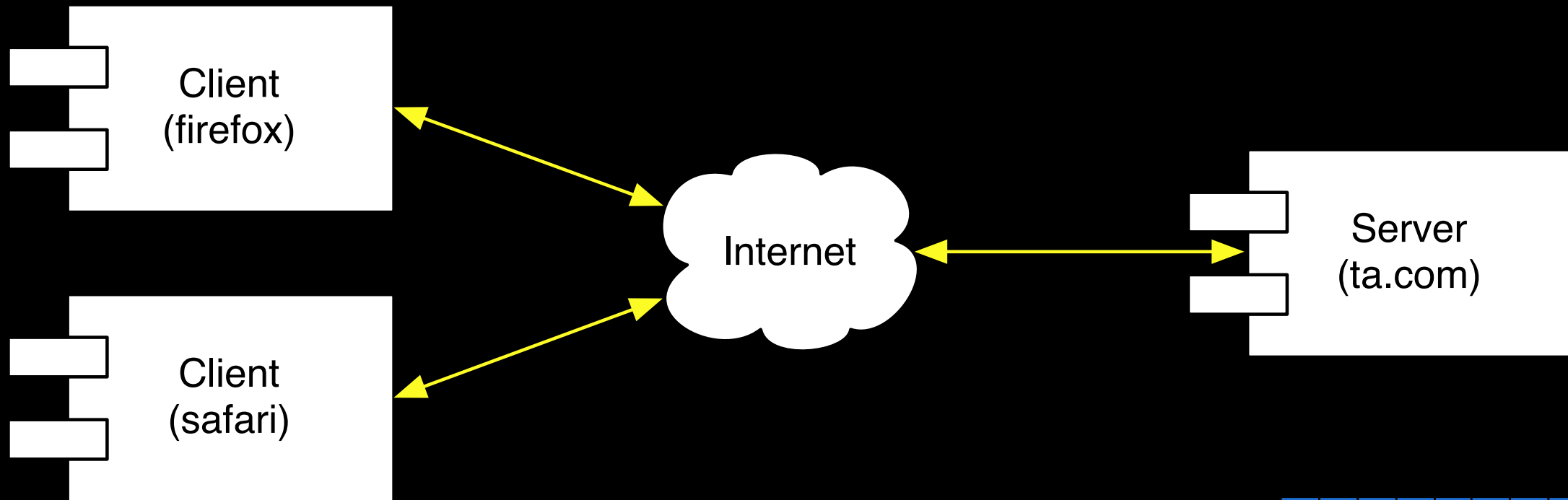
pauloborba.cin.ufpe.br

Software as a Service (SaaS)

- Delivers software and data as a service over the Internet
- No need to install applications and backup data in the user computing device
- Easier to improve the service

Client-server structure and behavior





Teaching Assistant

Alunos

Metas

Nome Adicionar

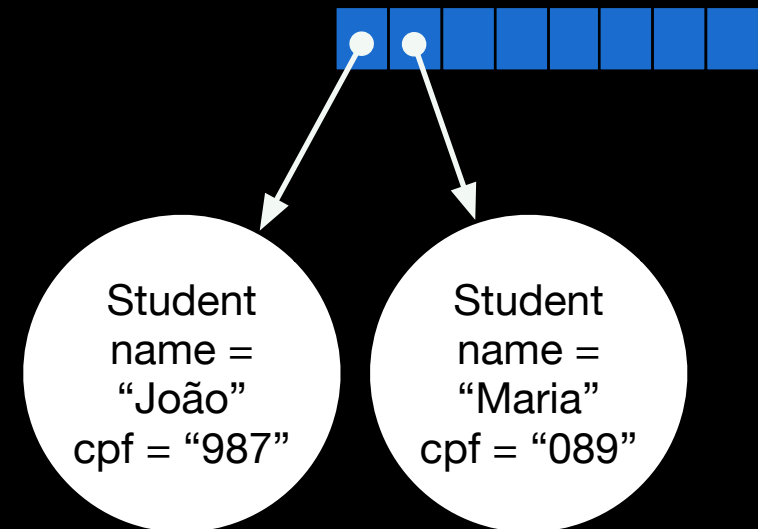
CPF

e-mail

Nome CPF e-mail

João 987 joao@cin.ufpe.br

Maria 089 maria@cin.ufpe.br



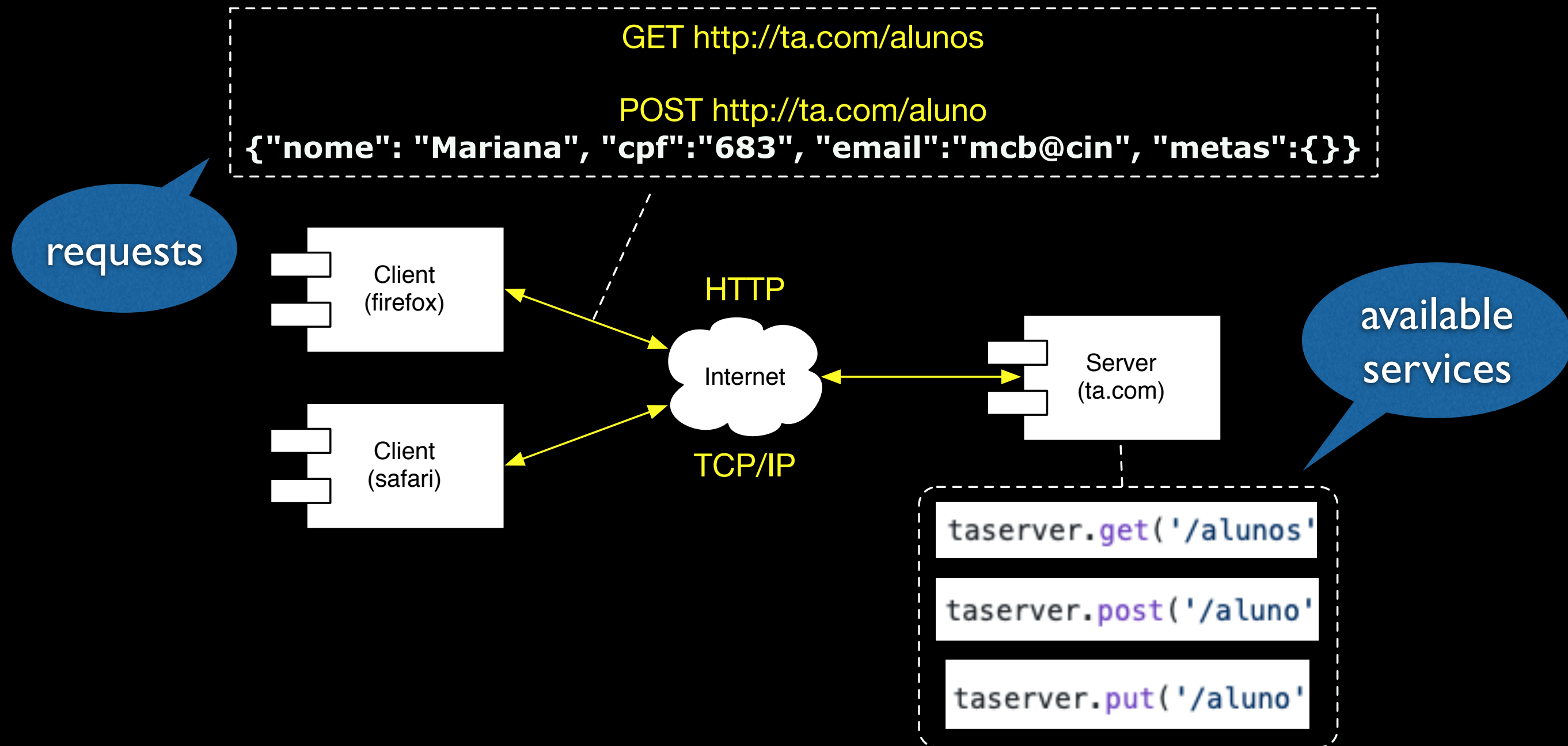
```

[ { "nome": "João", "cpf": "987",
    "email": "joao@cin.ufpe.br",
    "metas": { "requisitos": "MPA",
               "gerDeConfiguracao": "MA"
            }
  },
  { "nome": "Maria", "cpf": "089",
    "email": "maria@cin.ufpe.br",
    "metas": { "requisitos": "MA",
               "gerDeConfiguracao": "MA"
            }
  }
]
  
```

Pattern principles

- Separation of concerns between clients and servers
- Separates functionality and processing place
- Different interface for each server

Client-server communication protocols

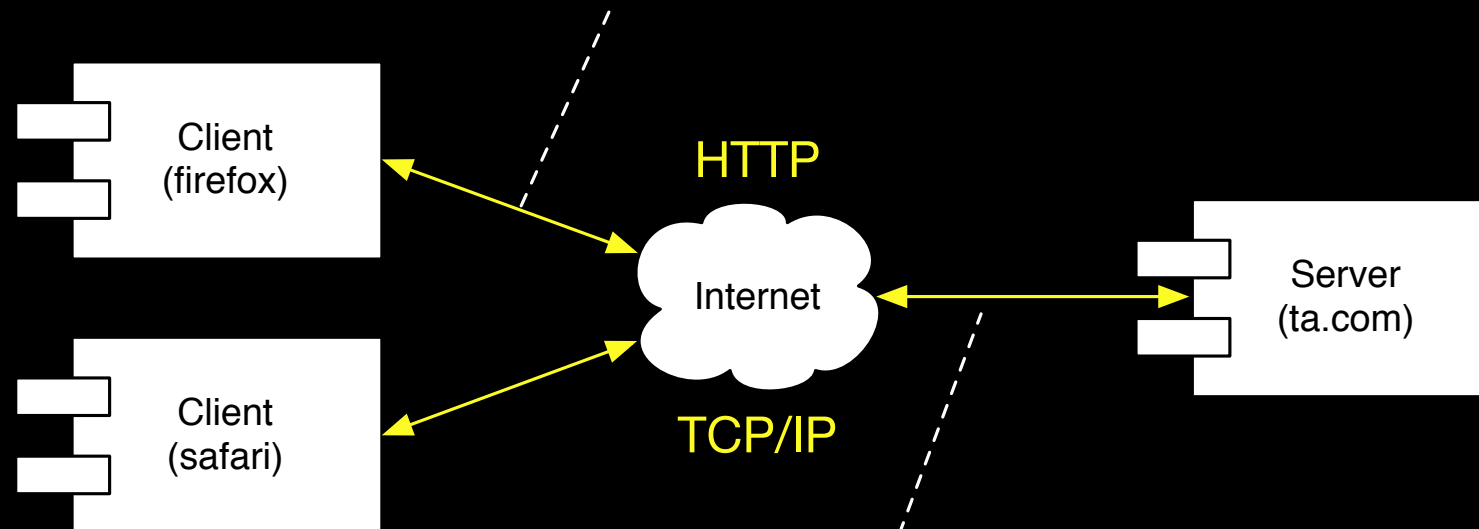


request

GET http://ta.com/alunos

POST http://ta.com/aluno

{"nome": "Mariana", "cpf": "683", "email": "mcb@cin", "metas": {}}



response

```
[ { "nome": "João", "cpf": "987",  
  "email": "joao@cin.ufpe.br",  
  "metas": { "requisitos": "MPA",  
             "gerDeConfiguracao": "MA"  
            }  
},  
  { "nome": "Maria", "cpf": "089",  
    "email": "maria@cin.ufpe.br",  
    "metas": { "requisitos": "MA",  
               "gerDeConfiguracao": "MA"  
            }  
}  
]
```

JSON
object

HTML + CSS + JS

(rendered by the browsers)

Teaching Assistant

Alunos Metas

Nome Adicionar

CPF

e-mail

| Nome | CPF | e-mail |
|-------|-----|-------------------|
| João | 987 | joao@cin.ufpe.br |
| Maria | 089 | maria@cin.ufpe.br |

```
<table>
<tr>
  <td><label>Nome </label></td>
  <td><input [(ngModel)]="aluno.nome" name="namebox"></td>
  <td><button (click)="criarAluno(aluno)">Adicionar</button></td>
</tr>
<tr>
  <td><label>CPF </label></td>
  <td><input [(ngModel)]="aluno.cpf" name="cpfbox"></td>
  <td name="msgcpfexistencia" >
    Já existe um aluno com e
  </td>
</tr>
<tr>
  <td><label>e-mail </label></td>
  <td><input [(ngModel)]="aluno.email" ></td>
</tr>
</table>
```

allows
users to request
services

shows
information to
users

allows users to
provide information

HTML

(specifies content to be rendered)

page,
element
hierarchy

element
tree

```
<table>
<tr>
  <td><label>Nome </label></td>
  <td><input [(ngModel)]="aluno.nome" name="namebox"></td>
  <td><button (click)="criarAluno(aluno)">Adicionar</button></td>
</tr>
<tr>
  <td><label>CPF </label></td>
  <td><input [(ngModel)]="aluno.cpf" name="cpfbox"></td>
  <td name="msgcpfexistente" *ngIf="cpfduplicado">
    Já existe um aluno com esse CPF
  </td>
</tr>
<tr>
  <td><label>e-mail </label></td>
  <td><input [(ngModel)]="aluno.email" ></td>
</tr>
</table>
```

value

element,
beginning

content

attribute, for
reference, or rendering and
event semantics

element,
end

tag, specifies what to render

HTML templates

(specifies how to create content from data)

```
<tr>
  <td><label>CPF </label></td>
  <td><input [(ngModel)]="aluno.cpf" name="cpfbox"></td>
  <td name="msgcpfexistente" *ngIf="cpfduplicado">
    Já existe um aluno com esse CPF
  </td>
</tr>
```

conditional
content

Teaching Assistant

Alunos

Metas

Nome

Adicionar

CPF

e-mail

Nome CPF e-mail

João 987 joao@cin.ufpe.br

Maria 089 maria@cin.ufpe.br

```
<table>
<tr *ngIf="alunos.length > 0">
  <th>Nome</th>
  <th>CPF</th>
  <th>e-mail</th>
</tr>
<tr *ngFor="let a of alunos" name="alunolist">
  <td name="nomelist">{{a.nome}}</td>
  <td name="cpflist">{{a.cpf}}</td>
  <td>{{a.email}}</td>
</tr>
</table>
```

content
generation loop

```

<table>
<tr *ngIf="alunos.length > 0">
  <th>Nome</th>
  <th>CPF</th>
  <th>e-mail</th>
</tr>
<tr *ngFor="let a of alunos" name="alunolist">
  <td name="nomelist">{{a.nome}}</td>
  <td name="cpflist">{{a.cpf}}</td>
  <td>{{a.email}}</td>
</tr>
</table>

```

HTML template,
written by a developer,
interpreted by a web framework

DOM object in the browser,
representing HTML obtained by
instantiating the template

Teaching Assistant

Alunos

Metas

Nome

Adicionar

CPF

e-mail

Nome CPF

e-mail

João 987 joao@cin.ufpe.br

Maria 089 maria@cin.ufpe.br

```

▶ <table _ngcontent-vxl-c1>...</table>
▼ <table _ngcontent-vxl-c1>
  <!--bindings={ "ng-reflect-ng-if": "true" }-->
  ▼ <tr _ngcontent-vxl-c1>
    <th _ngcontent-vxl-c1>Nome</th>
    <th _ngcontent-vxl-c1>CPF</th>
    <th _ngcontent-vxl-c1>e-mail</th>
  </tr>
  <!--bindings={ "ng-reflect-ng-for-of": "[object Object],[object Object]" }-->
  ▼ <tr _ngcontent-vxl-c1 name="alunolist">
    <td _ngcontent-vxl-c1 name="nomelist">João</td>
    <td _ngcontent-vxl-c1 name="cpflist">987</td>
    <td _ngcontent-vxl-c1>joao@cin.ufpe.br</td>
  </tr>
  ▼ <tr _ngcontent-vxl-c1 name="alunolist">
    <td _ngcontent-vxl-c1 name="nomelist">Maria</td>
    <td _ngcontent-vxl-c1 name="cpflist">089</td>
    <td _ngcontent-vxl-c1>maria@cin.ufpe.br</td>
  </tr>
</table>

```

html > body > app-root > html > app-root > html > table

rendering of the DOM object
in the browser

CSS (specifies presentation details of the content to be rendered)

| Nome | CPF | e-mail | Requisitos | Gerência de Configuração |
|-------|-----|-------------------|------------|--------------------------|
| João | 987 | joao@cin.ufpe.br | MPA | MA |
| Maria | 089 | maria@cin.ufpe.br | MA | MA |

CSS

```
<table>
  <tr>
    <th>Nome</th>
    <th>CPF</th>
    <th>e-mail</th>
    <th>Requisitos</th>
    <th>Gerência de Configuração</th>
  </tr>
  <tr *ngFor="let a of alunos">
    <td>{{a.nome}}</td>
    <td>{{a.cpf}}</td>
    <td>{{a.email}}</td>
    <td class="meta"><input [(ngModel)]="a.metas['requisitos']" (change)="atualizarAluno(a)"></td>
    <td class="meta"><input [(ngModel)]="a.metas['gerDeConfiguracao']" (change)="atualizarAluno(a)"></td>
  </tr>
</table>
```

HTML template

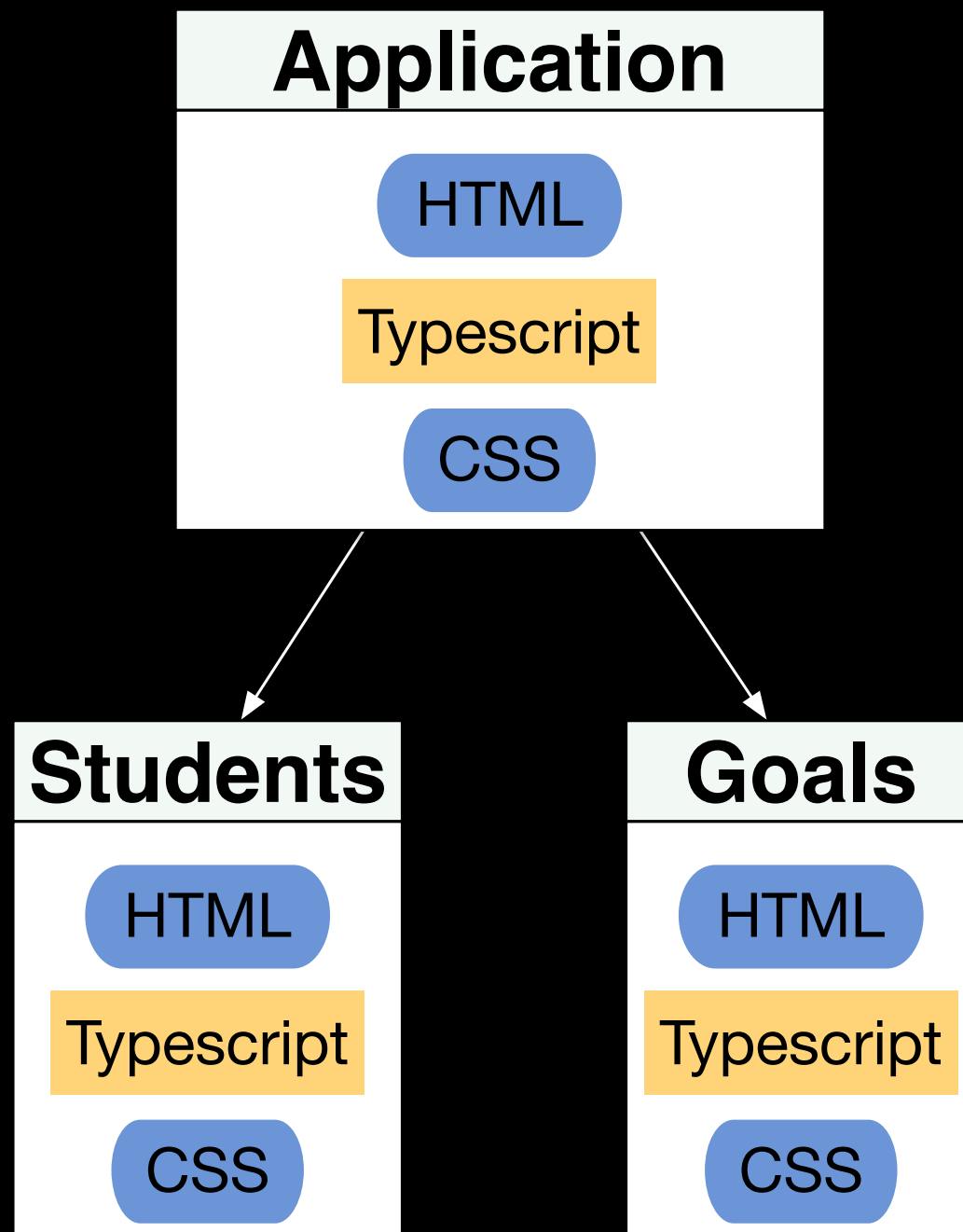
```
th {
  background-color: #3399ff;
  color: #ffffff;
}

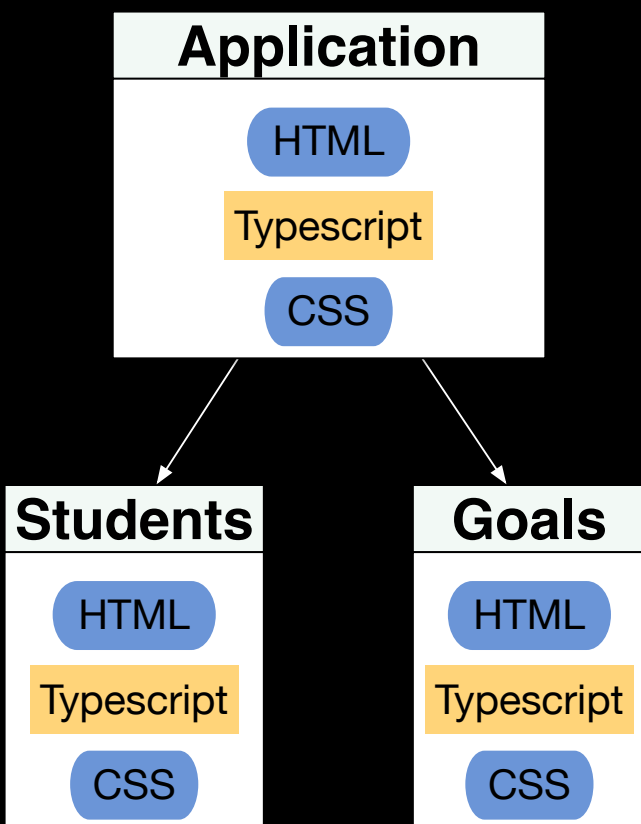
td, th {
  padding: 15px;
  border-bottom: 1px solid #ddd;
}

tr:hover {
  background-color: #cce6ff;
}
```

Web components

(modularizing client code)





Teaching Assistant

Alunos

Metas

| Nome | CPF | e-mail | Requisitos | Gerência de Configuração |
|-------|-----|-------------------|------------|--------------------------|
| João | 987 | joao@cin.ufpe.br | MPA | MA |
| Maria | 089 | maria@cin.ufpe.br | MA | MA |

Teaching Assistant

Alunos

Metas

Nome

Adicionar

CPF

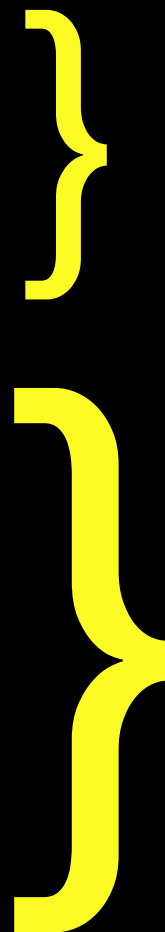
e-mail

Nome CPF

e-mail

João 987 joao@cin.ufpe.br

Maria 089 maria@cin.ufpe.br



Application

HTML

Typescript

CSS

Students

HTML

Typescript

CSS

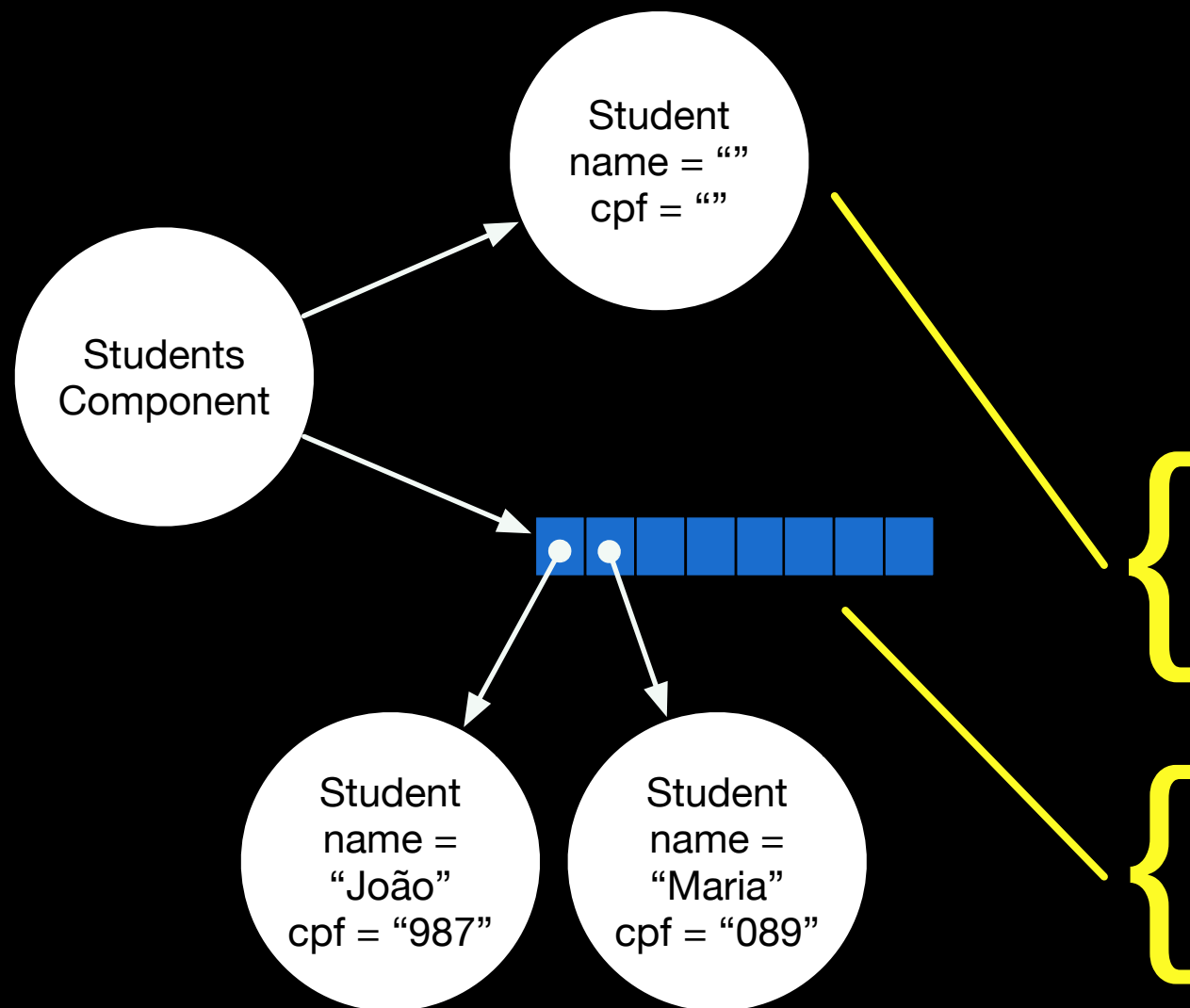
Goals

HTML

Typescript

CSS

JS (TS) objects as component state



Teaching Assistant

Alunos

Metas

Nome

Adicionar

CPF

e-mail

Nome CPF

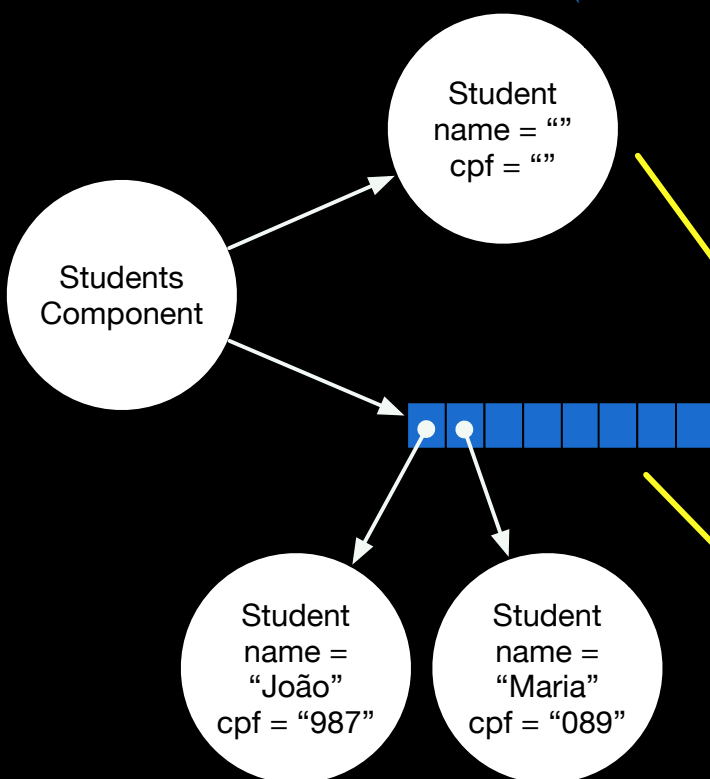
e-mail

João 987 joao@cin.ufpe.br

Maria 089 maria@cin.ufpe.br

state
read and written by
the interpretation of
the templates

specifies the
structure of the state,
and how it's initialised
and changed



Teaching Assistant

Alunos

Metas

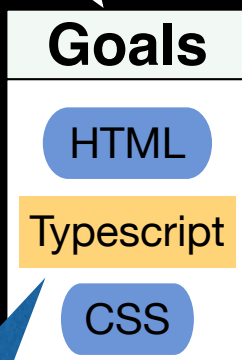
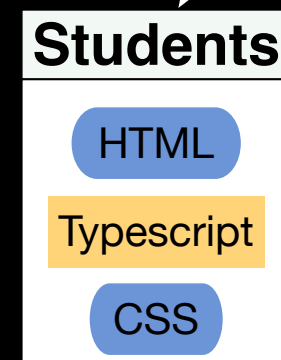
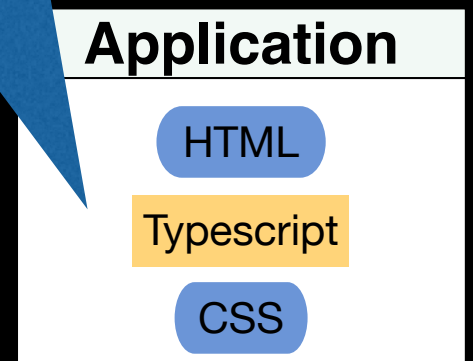
Nome

Adicionar

CPF

e-mail

| Nome | CPF | e-mail |
|-------|-----|-------------------|
| João | 987 | joao@cin.ufpe.br |
| Maria | 089 | maria@cin.ufpe.br |



result of the
rendering process by
interpreting the CSS and
HTML

specifies
operations triggered by
browser events

Angular architecture

- Module contain
- Components
 - contain HTML, CSS, Typescript class
 - use services (Typescript class)
 - refer to models (Typescript class)

Take notes,
now!

Hands on exercises

SaaS I

SaaS 2

For each commit...

- Check system running
- Check commit changes
- Starting at second commit and going up to “melhor visualizacao do botao e da lista de alunos”

Hands on exercises

SaaS 2

SaaS 3

For each commit...

- Check system running
- Check commit changes
- Starting at commit "evitar cadastro de mais de um aluno com o mesmo cpf" and going up to “melhorias na visualizacao de metas”

Hands on exercises

SaaS 3

SaaS 4

For each commit...

- Check system running
- Check commit changes
- Starting at commit "versao inicial do servidor" and going up to first part of "servidor integrado ao cliente"

Hands on exercises

SaaS 4

SaaS 5

For each commit...

- Check system running
- Check commit changes
- Starting at the second part of commit
“servidor integrado ao cliente”, going up to
"tratamento de erro do atualizar”

Hands on exercises

SaaS 5

SaaS 6

Hands on exercises

SaaS 6

SaaS research at CIn

- Cloud and distributed architectures: Kiev, Vinicius, Castor
- Specification of distributed architectures: Alexandre Mota, Augusto

To do after class

- Answer questionnaire (check classroom assignment), study correct answers
- Finish exercise (check classroom assignment), study correct answers
- Read, again, parts of chapters 7 and 10 in the textbook
- Evaluate classes
- Study questions from previous exams

To do after class

- HTML: <https://www.w3schools.com/tags/>
- CSS: <https://www.w3schools.com/css/default.asp> (reference), https://www.w3schools.com/colors/colors_picker.asp (colors)
- Angular: <https://angular.io/guide/> (reference), <https://stackblitz.com> (IDE), <https://plnkr.co> (IDE), <https://angular.io/tutorial> (tutorial)
- Typescript: <https://www.typescriptlang.org/docs/home.html>
- Javascript: <https://www.w3schools.com/jsref/default.asp> (básico), <http://exploringjs.com/es6/index.html> (avançado)
- Node.js: <https://nodejs.org/en/>
- Express: <http://expressjs.com>
- body-parser: <https://www.npmjs.com/package/body-parser>
- RxJS: <http://reactivex.io/rxjs/> (API), <https://www.learnrxjs.io>

Questions from previous exams

- Explique brevemente a diferença entre testes de unidade e testes de integração (a). Qual o impacto negativo de realizar apenas os testes de unidade? (b) Qual o impacto negativo de realizar apenas os testes de integração?
- Explique brevemente a diferença entre testes de aceitação e testes de integração, e porque você acha que algumas empresas realizam os dois tipos de teste.

Software Engineering

Paulo Borba
Informatics Center
Federal University of Pernambuco

pauloborba.cin.ufpe.br