

Monica



Personal Relationship Management

Alexis SAETTLER

FOSDEM 2019



1. Presentation of Monica

2. What we are trying to do



1. Presentation of Monica

2. What we are trying to do

Idea



- Remember everything about your contacts
 - Personal information (family, age, habits, ...),
 - Activities,
 - Notes,
 - Etc.
- Aimed to those who lack memory



Where to store this data?

- At some big companies
 - Facebook, Google: it's a no.
- In a file (text or sheet editor)
 - Not queryable
 - Not optimized for this use case
- Another contact manager

The answer is Monica

What is it for?



- It is for the contacts you care about
For whom you want to remember things
→ About your **friends**, colleagues, and more.



What is it not?



- Not a social network
 - Data is only for yourself, they will never be shared with other users of one instance, nor marketing or ad services.
- Not a file hosting service
 - There are better software out there.
- Not a robot with some A.I.
 - Reminders are sent because you want it.

Features



- Manage contacts and relationships (address book)
- Birthdays
- Contact information: email, phone, address, etc.
- Tag contacts
- Avatar
- Favorite a contact

Features



- Add reminders for birthdays or to stay in touch
- Tasks
- Notes
- Activities or Life events
- Debts
- Gifts

Features



- Geolocalize addresses
- Weather information
- Upload documents and photos
- Export as a vcard
- Conversations from social media or SMS (not emails)
- Phone calls
- Personal journal

Features



- REST Api
- 15 locales
- Multi-Factor authentication (2FA/U2F)
- Import/Export



Search your contacts...

Dashboard

Contacts

Journal



[Dashboard](#) > [List of people](#) > John Doe



John Doe ★

Edit contact information



Feb 13, 1996 (22)



Last activity together: unknown



Last called: Jan 31, 2019



Stay in touch every day [Edit](#)

Best friend

[Edit](#)

LOVE RELATIONSHIPS

[Add](#)

FAMILY RELATIONSHIPS

sister

Jane Doe (28) [Edit](#) [Delete](#)

[Add](#)

OTHER KIND OF RELATIONSHIPS

[Add](#)

PETS

[Add](#)

Life events (0)

Notes, reminders, ...

Photos

Add note

Yeah

★ Sep 19, 2018

[Edit](#) | [Delete](#)



Conversations

Log conversation

Record conversations you have with John on social media, SMS, ...

[Log conversation](#)

Still some work to do !



- Better UX
- CardDAV and CalDAV
 - Almost there, we need feedback !
- Some refactorization
- Notion of groups
- Notion of « me »
- Security
 - End-to-end encryption

The mobile application



- Huge demand by all users
- We used to have a mobile app called **Chandler** but its maintainer is no longer in the project
- Now we need a super skilled person to continue this work.



How to use it



- Our hosted version on **app.monicaHQ.com**
 - No user tracking (no Google Analytics or facebook widget)
- Or install it, like 8000+ others
 - Your own server
 - From the sources
 - From Docker (~780K pulls)
 - YunoHost
 - Heroku
 - Etc.



1. Presentation of Monica

2. What we are trying to do

What it is made from



- Laravel framework
- Blade and Vue.js pages
- Css
- Love and passion

An Open Source app



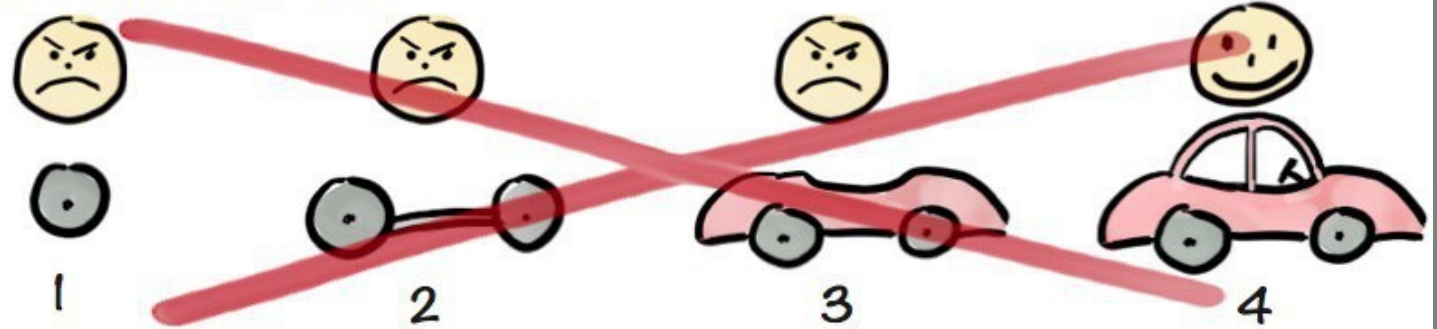
- Most voted story on Hacker news in 2017
- Most php trending repo on github for 2 months (2018)
 - 6500 stars on github
- Open source Rookie of the Year 2018
- +140 contributors
- ~27K lines of code

Our development process model

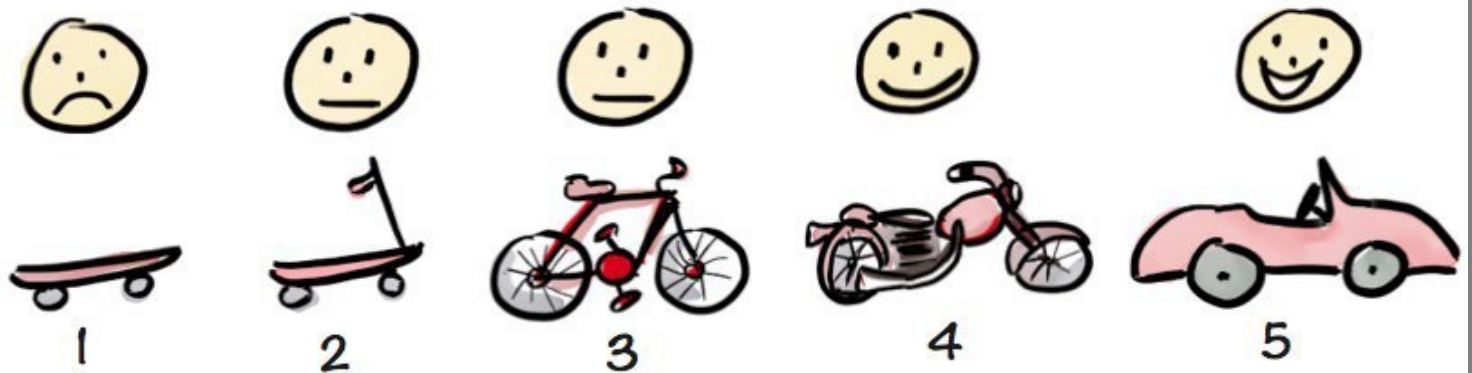


- Ship little features, often
- Improve it over multiple iterations

Not like this....



Like this!



Work on an Open Source project



- A lot of github issues
 - Lesson: don't agree to everything, do what **you** want to do with your product
- Accept all Pull requests ?
 - Yes / no
 - Every update on the code will be babysitted by the owner
 - Some times we have to reject a pull request, but it's heartbreaking



What we are trying to do

1. An application for everyone
 - Nice UI, UX
 - Support every locales
2. An application fully tested (well, almost fully)
 - With a nice CI
3. An application with which we can have fun in the development



2.1 An application for everyone

2.2 A fully tested application

2.3 An application we have fun developing

An application for everyone



Simple

Fast

Reliable

An application for everyone



- Everyone means everywhere
- We support 15 locales, with country specificities
 - ar cs de en es fr he hr it nl pt pt-BR ru tr zh

Internationalisation



- Source language: English
- Translation with an external tool
 - Crowdin has been really helpful for us!
- Special attention to
 - Localise everything!
 - Lang specificity: date format, money value, plural forms
 - Country specificity: temperature scale, metric system
 - Collation (sorting)
 - Direction: left-to-right/right-to-left



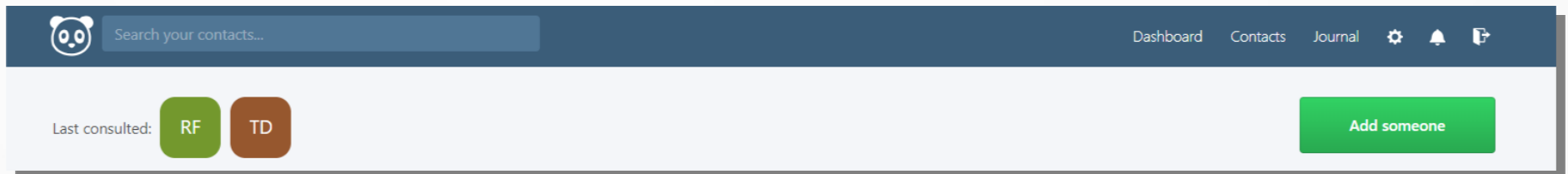
Problem with ltr/rtl

- Adding html "dir" attribute not solves everything:

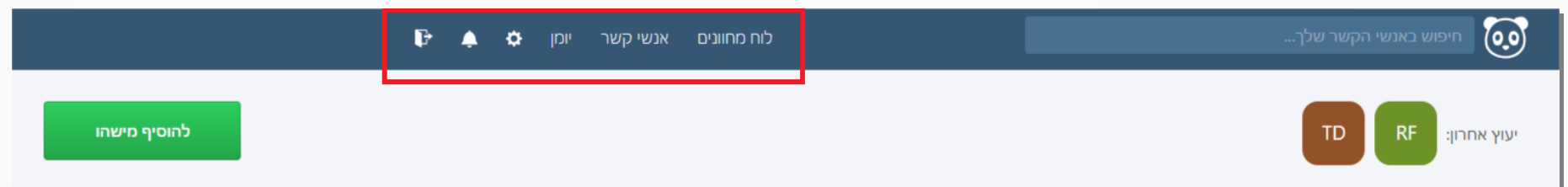
```
<!DOCTYPE html>
<html lang="he" dir="rtl">
```

- Example:

- English



- Hebrew



- Reason:

```
.header-nav {
  text-align: right;
```



How to handle ltr/rtl

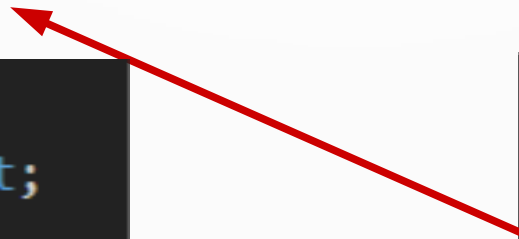
- 1. Detect language and direction
 - 12 languages with rtl
- 2. Two css files

app-ltr.scss

```
...  
$htmlDir: ltr !default;  
...
```

app-rtl.scss

```
$htmlDir: rtl;  
  
@import "app-ltr";
```





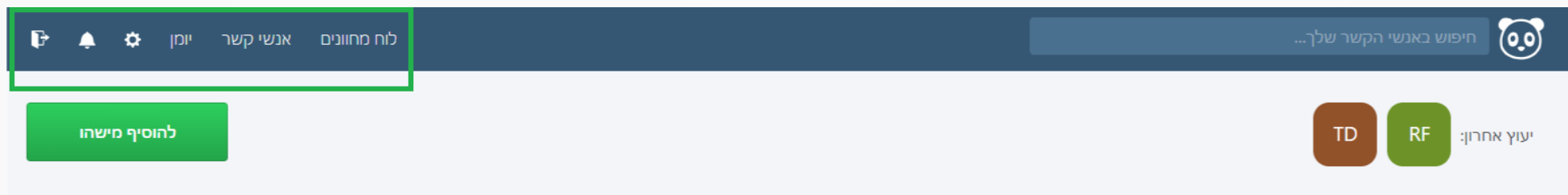
How to handle ltr/rtl

- 3. Use `$htmlDir` prop

```
.header-nav {  
  @if $htmlDir == ltr {  
    text-align: right;  
  } @else {  
    text-align: left;  
  }  
}
```

- 4. Use right css file

```
<link rel="stylesheet" href="{{ asset(mix('css/app-' .htmlDir() . '.css')) }}">
```





2.1 An application for everyone

2.2 A fully tested application

2.3 An application we have fun developing

An application fully tested



What you need:

1. Sources
2. Tests
3. Run tests with code coverage
4. Reporting



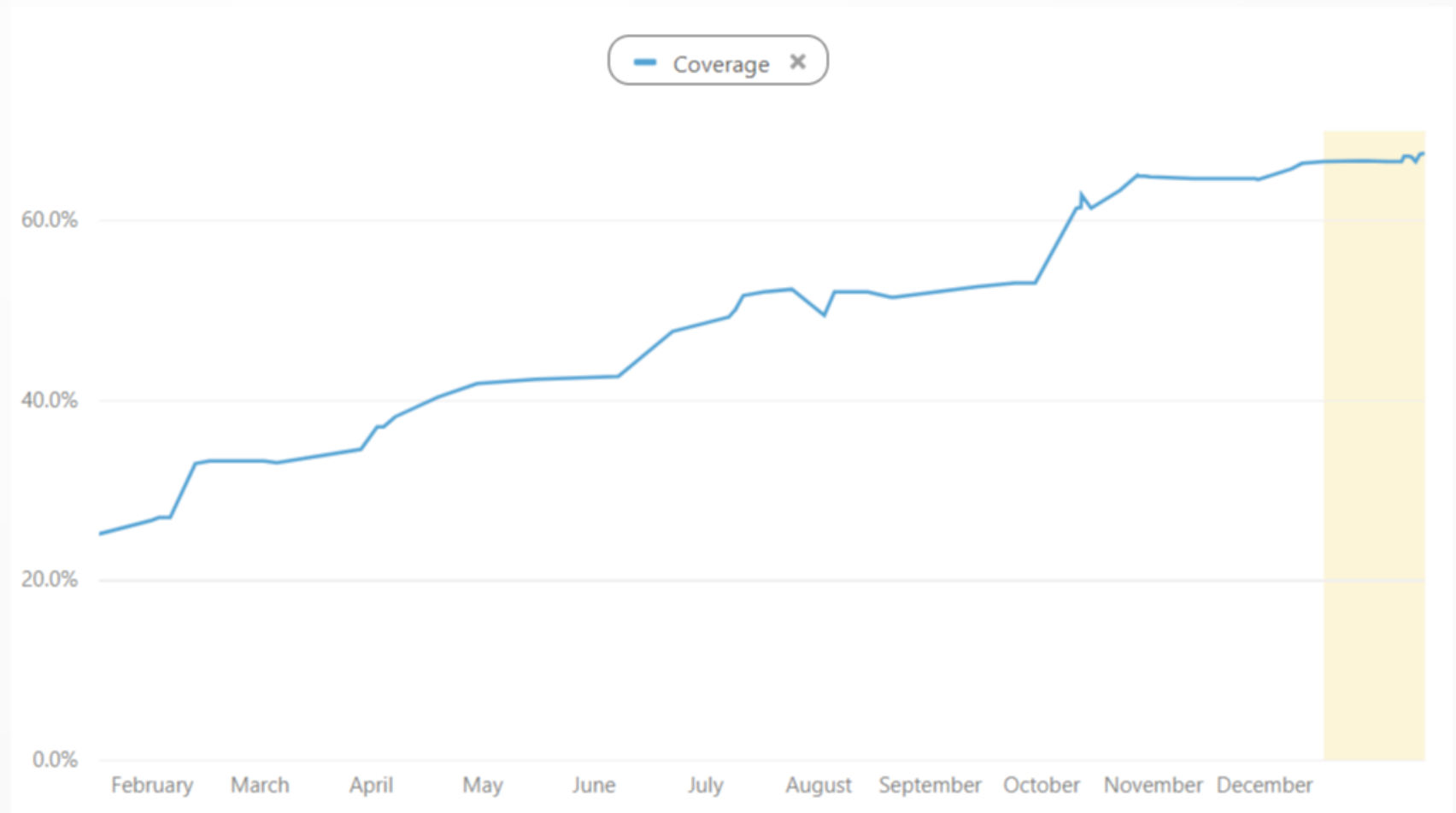
An application fully tested

1. Sources → Github
2. Tests → Unit/Browser tests
3. Run tests with code coverage
 - Continuous Integration: CircleCI
 - Unit tests: phpunit + phpdbg
 - Browser test: Laravel Dusk + "live code coverage"
4. Reporting
 - Sonarcloud

Current coverage



- ~27K lines of code
- 68% php code coverage



Our current CI

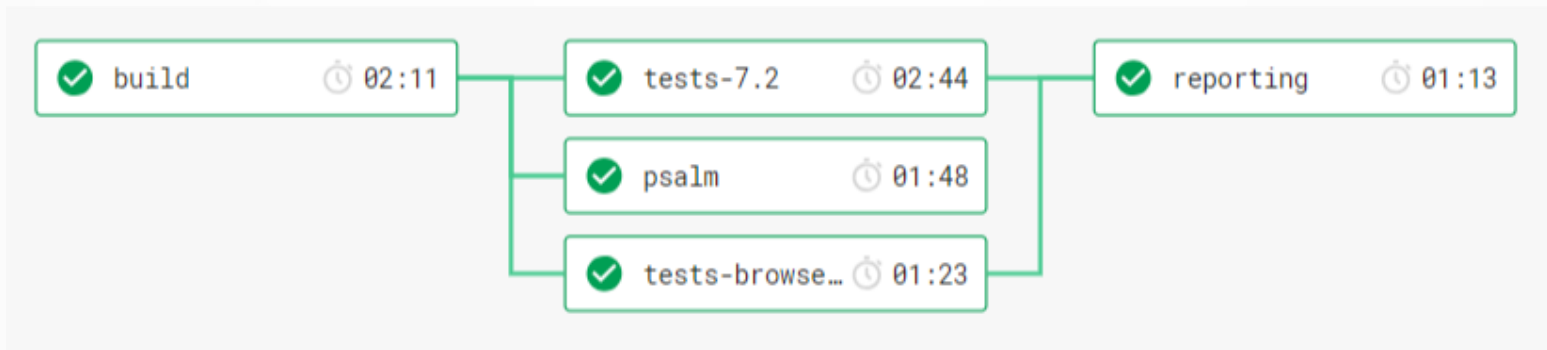


- Formerly TravisCI
- Now CircleCI
 - Workflow is great !
 - Using docker container is stable
 - We even created our own docker image: **centralperk**

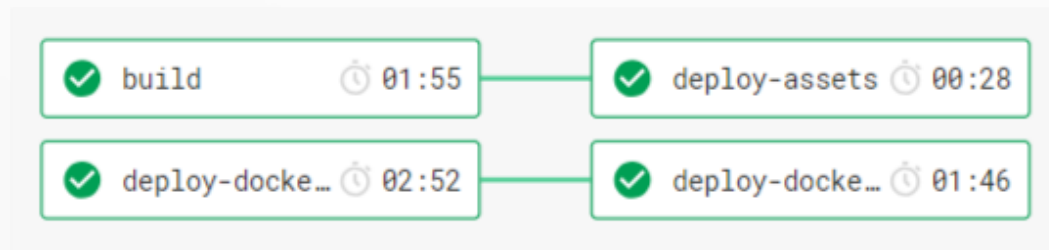
CircleCI workflow



- Build and test workflow



- Deploy workflow (master)





Tests : gain some time

- Before
 - Personal computer, no xdebug: ~2:30
 - On CI (phpdbg): ~4:20

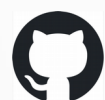
- Optimisation, from Taylor Otwell

Add this on CreatesApplication trait:

```
// set the bcrypt hashing rounds (the minimum allowed).  
// this reduces the amount of cycles needed to manage users.  
Hash::setRounds(4);
```

- Personal computer, no xdebug: ~1:40
 - On CI (phpdbg): ~3:00 → **30% faster !** 👍

CI



Github



CircleCI



StyleCI



Sonarcloud



Bintray



Docker hub



→ Long live OpenSource !



2.1 An application for everyone

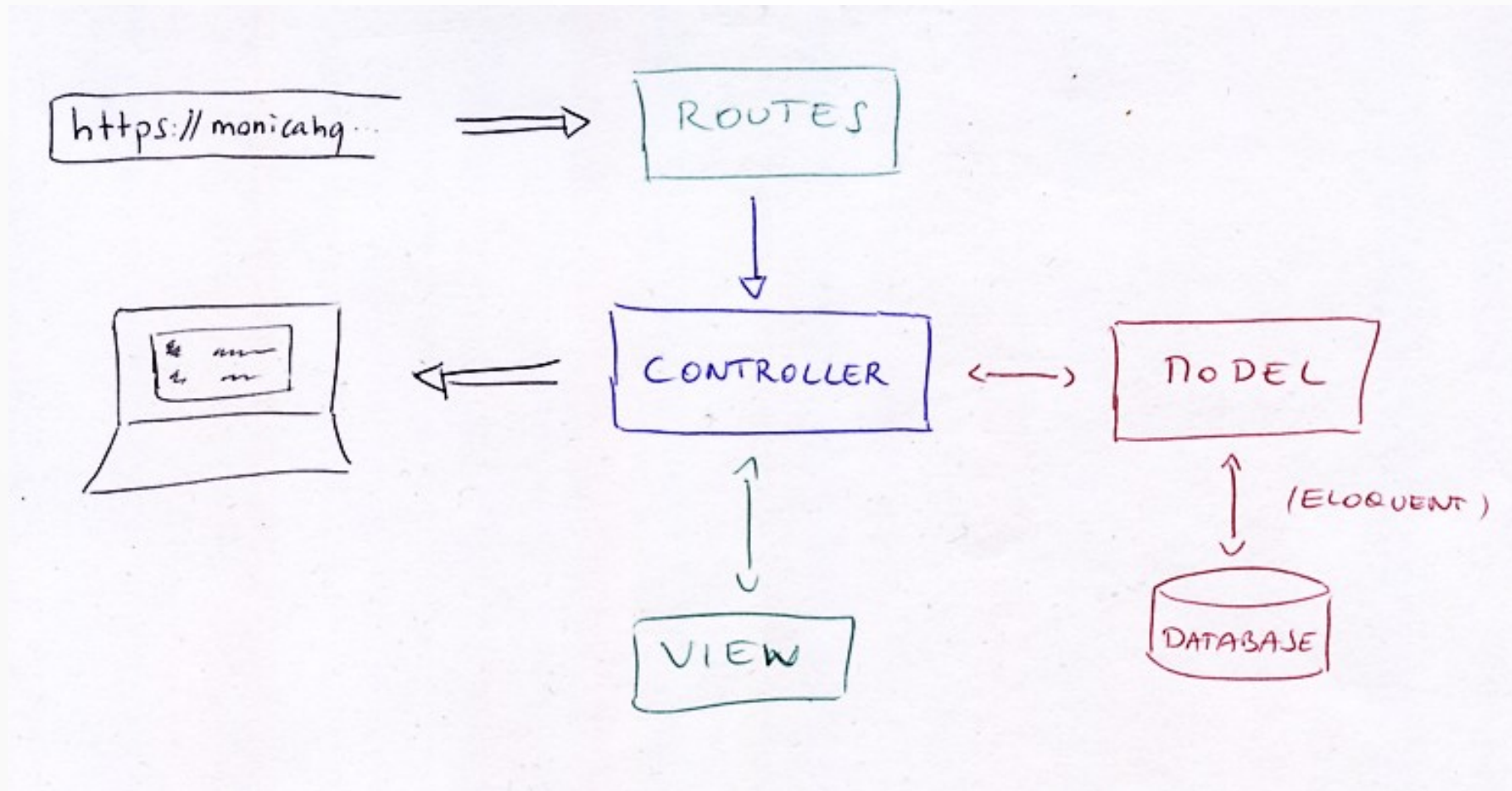
2.2 An application fully tested

2.3 An application we have fun developing

Architectural structure



- Laravel application



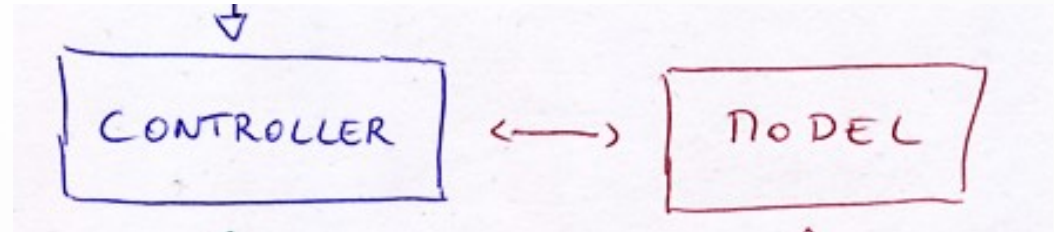


Introducing Services

- Use case:

- View Controller
- Api Controller

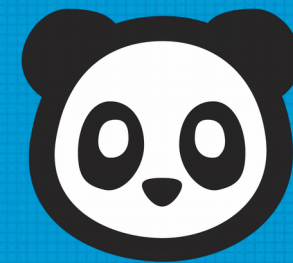
→ they need the same code to call the CRUD operations



- Solution: using **services**

- logic is handle by the service
- easier to maintain

Service : Example



```
class CreateAddress extends BaseService
{
    /**
     * Create an address.
     *
     * @param array $data
     * @return Address
     */
    public function execute(array $data) : Address
    {
        $this->validate($data);

        Contact::where('account_id', $data['account_id'])
            ->findOrFail($data['contact_id']);

        $place = $this->createPlace($data);

        return Address::create([
            'account_id' => $data['account_id'],
            'contact_id' => $data['contact_id'],
            'place_id' => $place->id,
            'name' => $this->nullOrValue($data, 'name'),
        ]);
    }
}
```

Service : View Controller



```
/**
 * Store the address.
 */
public function store(Request $request, Contact $contact)
{
    $datas = [
        'account_id' => auth()->user()->account_id,
        'contact_id' => $contact->id,
    ] + $request->only([
        'name',
        'country',
        'street',
        'city',
        'province',
        'postal_code',
        'latitude',
        'longitude',
    ]);

    $address = app(CreateAddress::class)->execute($datas);

    return $this->setHTTPStatusCode(201)
        ->respond($this->addressObject($address));
}
```


Service : Api Controller



```
/**
 * Store the address.
 *
 * @param Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    try {
        $address = app(CreateAddress::class)->execute(
            $request->all()
            +
            [
                'account_id' => auth()->user()->account->id,
            ]
        );
    } catch (ModelNotFoundException $e) {
        return $this->respondNotFound();
    } catch (ValidationException $e) {
        return $this->respondValidatorFailed($e->validator);
    } catch (QueryException $e) {
        return $this->respondInvalidQuery();
    }

    return new AddressResource($address);
}
```


You want to help ?



- We always need arms and brains
 - Code
 - Discuss features
 - UX
 - I18n (on Monica's crowdin project)
 - Mobile application



monicahq.com

Alexis Saettler @asbin

Régis Freyd @djaiss