

Refine Online Analysis Web with Laravel

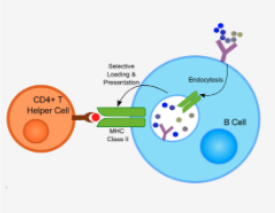
by Fangjiang Wu

Background

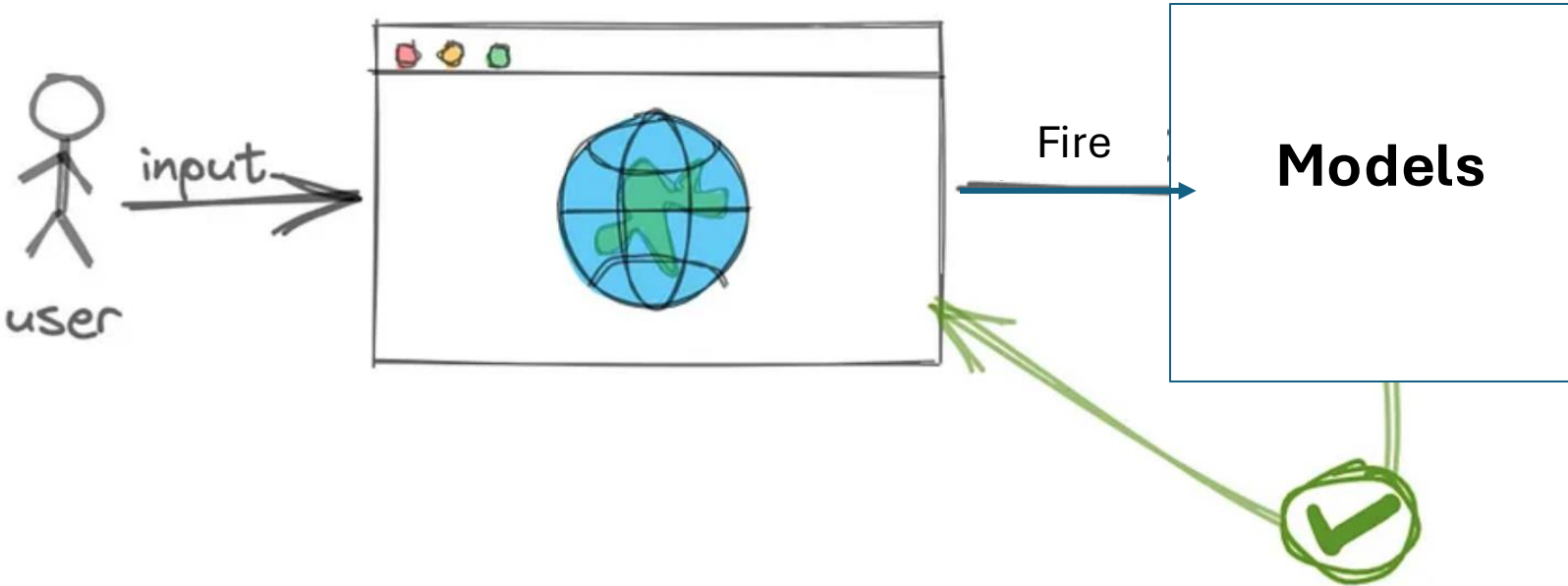
Analysis web tools



pMTnet



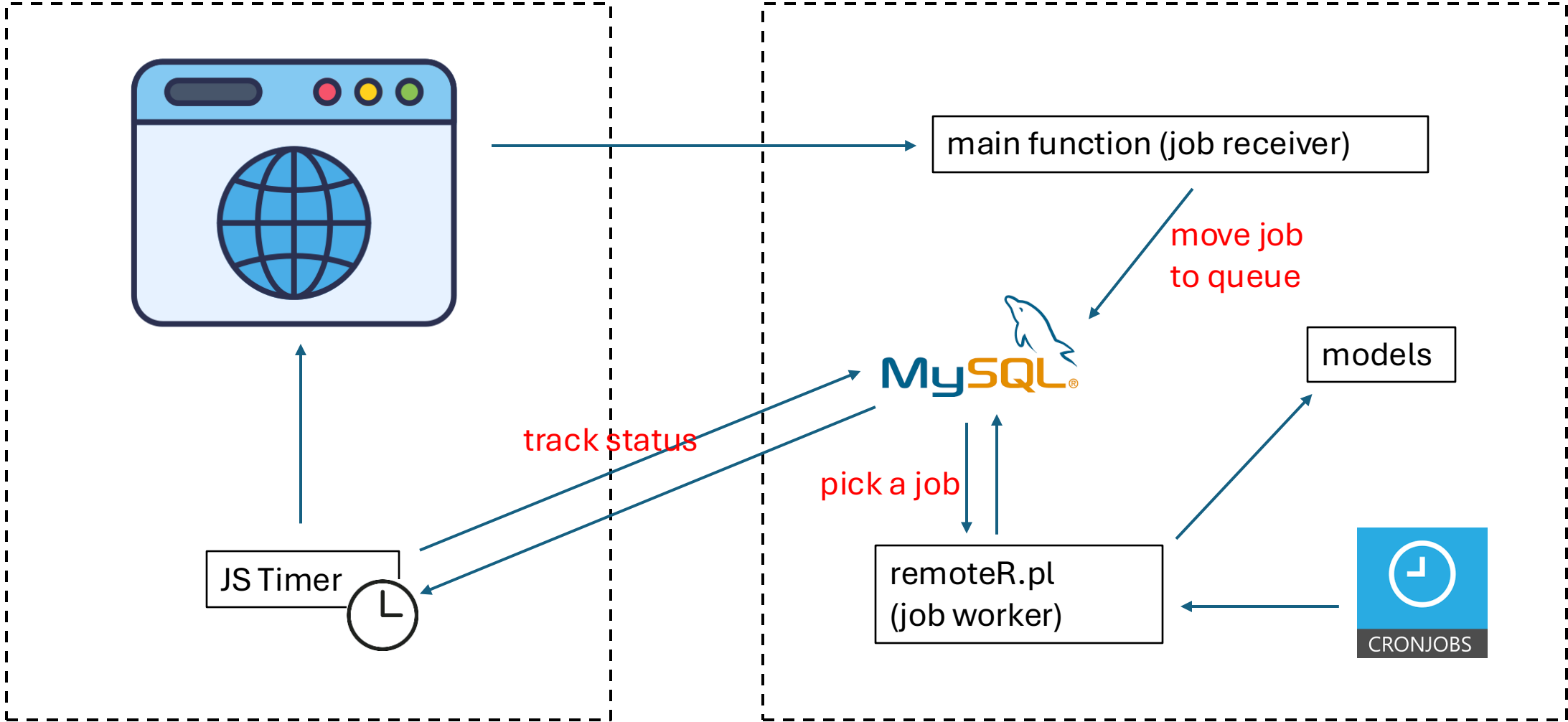
BepiTBR



Background - RemoteR

Frontend

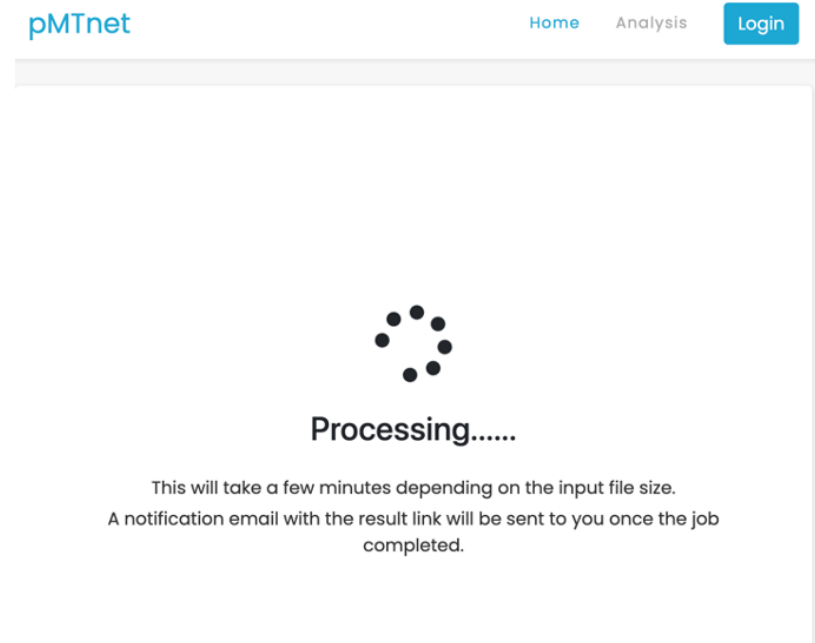
Backend



Background - RemoteR

Drawbacks:

1. Mixed up with **3** different languages (PHP, Perl, Bash Script)
 - Global setting in crontab
 - Parameters passing
 - Access permissions
2. Lack of Extensibility
3. Loading spinner is a UX killer!



Laravel

What is Laravel?

Laravel is a free and open-source PHP web framework.



Powerful Features

1. Job Queue

- simplify the process of job dispatching and tracking

2. Broadcasting (WebSocket connection)

- live update information at frontend

Laravel

Join future analysis tools with DBAI



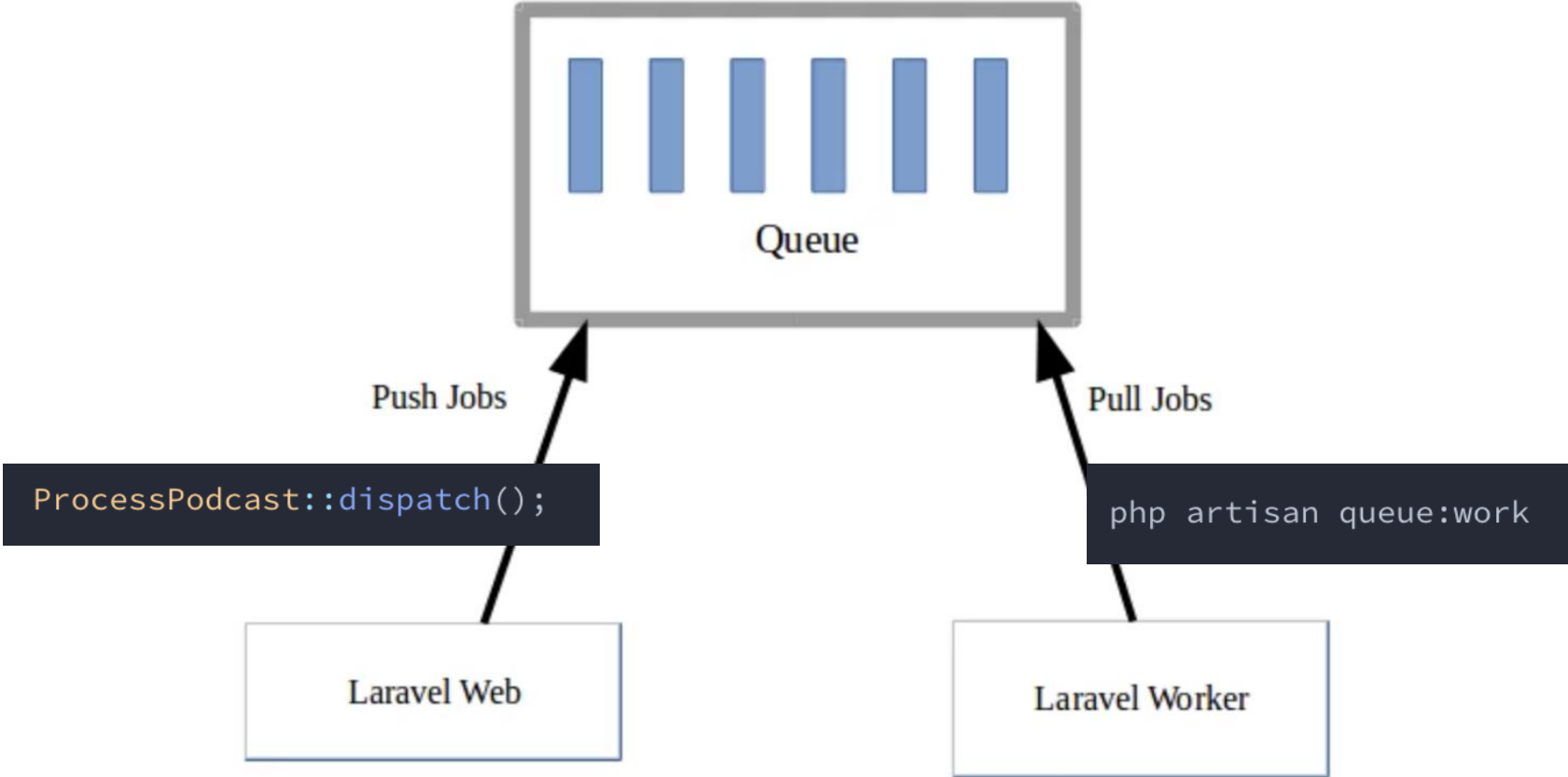
The screenshot shows the dbAI website interface. The logo at the top left reads "dbAI The Database of Actionable Immunology". The navigation menu includes "Home", "Database", "Tools", and a "Login" button. The main content area is titled "Databases" and lists several database categories, each with a red database icon and a record count:

Database	Records
Receptor	366
ScRNA	0
Individual	120
Sample	270
T interaction	0
B interaction	0
CyTOF	12

Laravel – Job Queue

+ Options

	id	queue	payload	reserved_at	available_at	attempts	created_at
<input type="checkbox"/>	82	default	{"uuid":"e5a3e4df-43c6-4e5b-bb46-5e31acce697","di...	1722229869	1722229868	1	1722229868



Laravel – Job Queue

Dispatch and run job in queue

```
public function dispatchJob(Request $req)
{
    dispatch(new RunModel());
    return back();
}
```

```
public function handle(): void
{
    $process = Process::forever()->start('/home/fangjiang/miniconda3/envs/omni/bin/python
-m pMTnet_Omni
--data_dir /home/fangjiang/public_html/pMTnet/omni/pMTnet_Omni/data
--user_data /home/fangjiang/public_html/pMTnet/omni/test_output/df_curated.csv
--output_file_path /home/fangjiang/public_html/pMTnet/omni/test_output/prediction.csv
--compute_percentile_rank
--check_size 50000
--load_size 1000000
--minibatch_size 10000');

    while ($process->running()) {
        $info = $process->latestOutput();
        JobStatusUpdated::dispatch('fangjiang', $info);
        sleep(5);
    }
}
```


Laravel – Job Queue

Queue worker

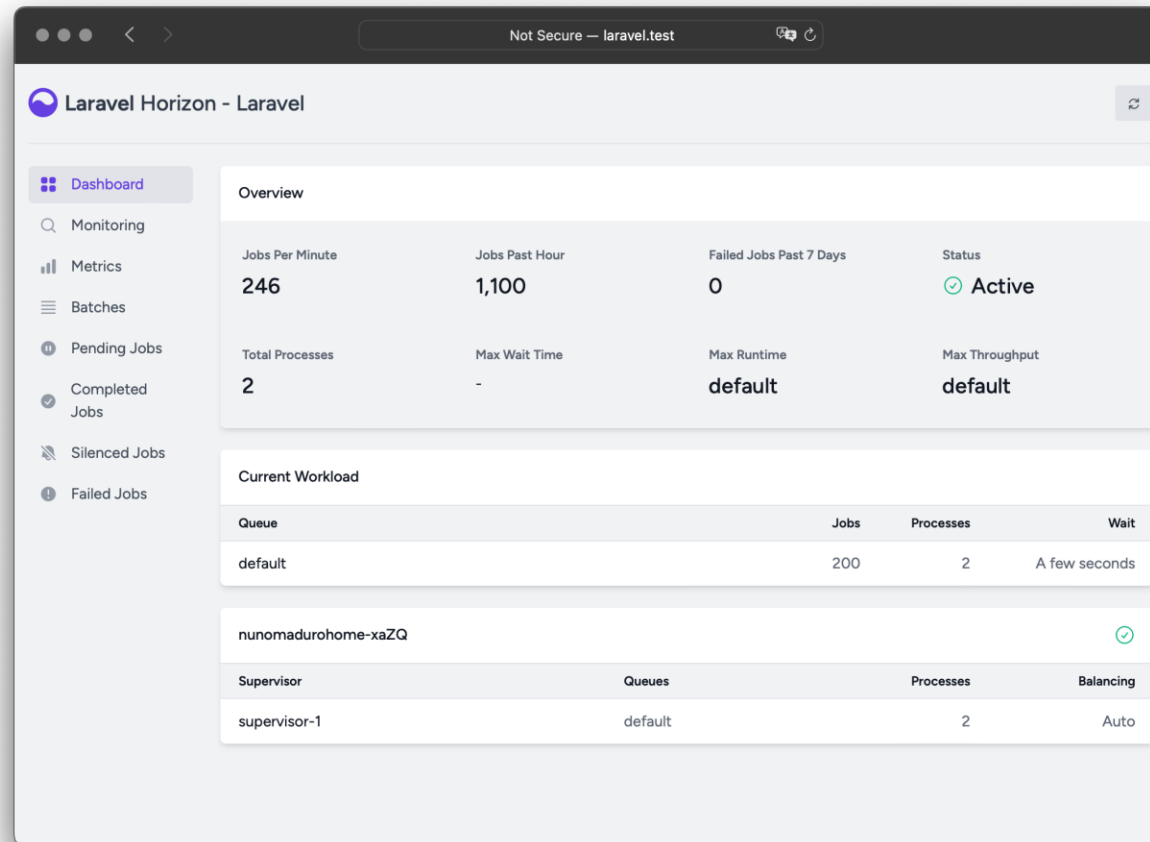
```
php artisan queue:work
```

No need to trigger job tracking script repeatedly

```
remoter_path=/home/fangjiang/remoter/  
web_path=/home/fangjiang/public_html/  
remoter_host=localhost  
remoter_usr=fangjiang  
remoter_passwd=[REDACTED]  
remoter_db=[REDACTED]  
remoter_lcedb=[REDACTED]  
* * * * * cd $remoter_path && perl runremoteR.pl >> perl.log 2>&1; cd /home/far  
* * * * * sleep 10; cd $remoter_path && perl runremoteR.pl >> perl.log 2>&1 &  
* * * * * sleep 20; cd $remoter_path && perl runremoteR.pl >> perl.log 2>&1 &  
* * * * * sleep 30; cd $remoter_path && perl runremoteR.pl >> perl.log 2>&1 &  
* * * * * sleep 40; cd $remoter_path && perl runremoteR.pl >> perl.log 2>&1 &  
* * * * * sleep 50; cd $remoter_path && perl runremoteR.pl >> perl.log 2>&1 &
```

Laravel – Job Queue

Laravel Horizon + Redis DB



The screenshot displays the Laravel Horizon dashboard for a Laravel application. The browser address bar shows 'Not Secure - laravel.test'. The dashboard title is 'Laravel Horizon - Laravel'. A left sidebar contains navigation links: Dashboard (selected), Monitoring, Metrics, Batches, Pending Jobs, Completed Jobs, Silenced Jobs, and Failed Jobs. The main content area is titled 'Overview' and features a grid of metrics: Jobs Per Minute (246), Jobs Past Hour (1,100), Failed Jobs Past 7 Days (0), and Status (Active with a green checkmark). Below this is a table for 'Current Workload' with columns for Queue, Jobs, Processes, and Wait. The 'default' queue is shown with 200 jobs and 2 processes, with a wait time of 'A few seconds'. A section for 'nynomadurohome-xaZQ' shows a green checkmark. At the bottom, a table lists supervisors: 'supervisor-1' is associated with the 'default' queue, has 2 processes, and is set to 'Auto' balancing.

Queue	Jobs	Processes	Wait
default	200	2	A few seconds

Supervisor	Queues	Processes	Balancing
supervisor-1	default	2	Auto

Laravel – Job Queue

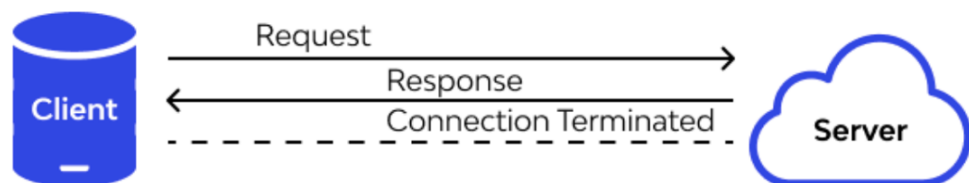
Some other useful features

1. Categorized queue with priorities
2. batch up jobs -> show progress bar frontend

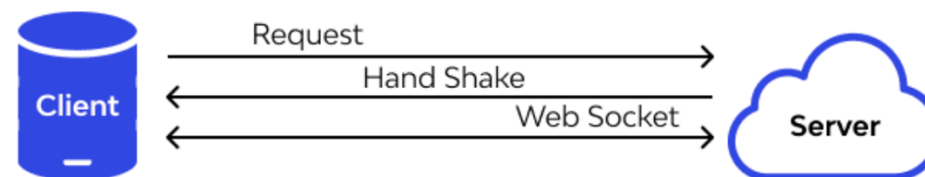
Laravel – Broadcasting

HTTP connection **vs** WebSocket connection

HTTP Connection

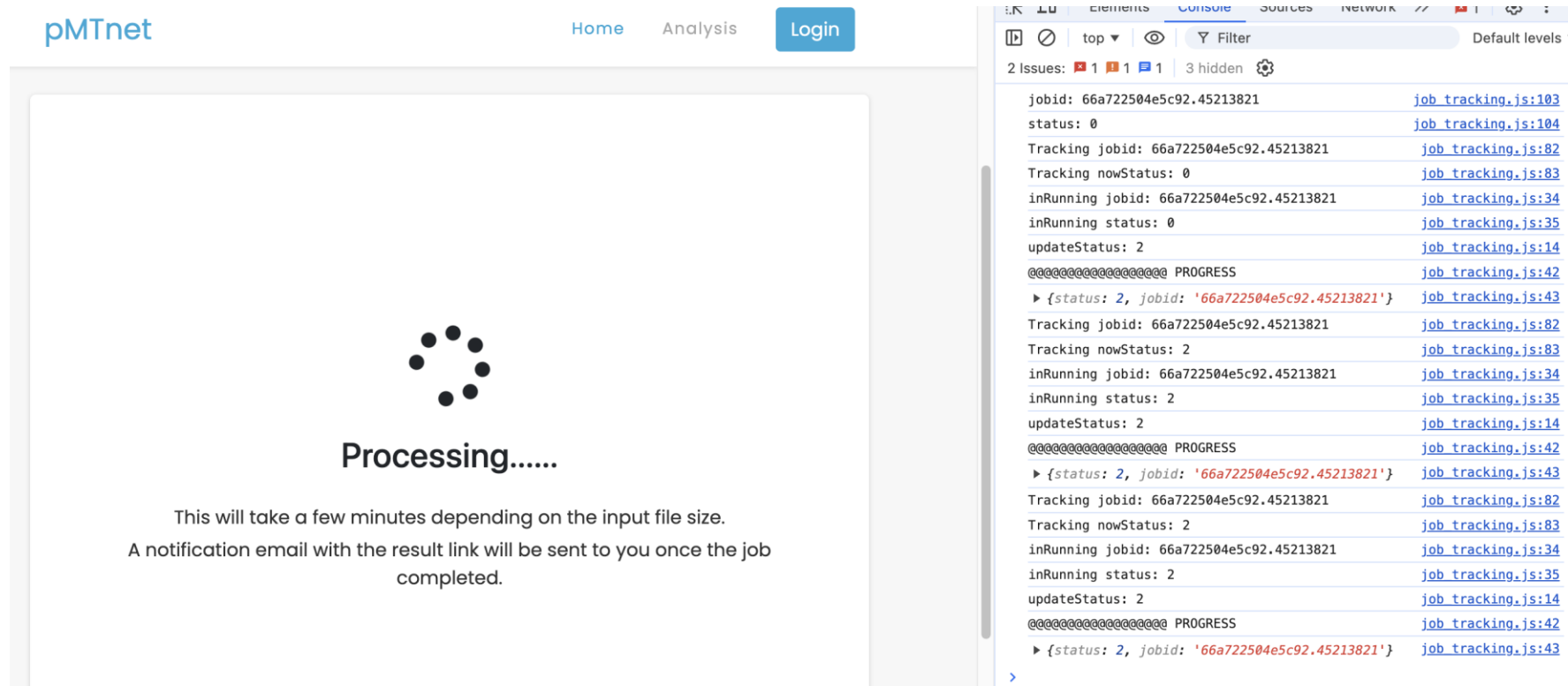


WebSocket Connection



Laravel – Broadcasting

HTTP connection in RemoteR



The screenshot shows a web application interface for pMTnet. The main content area displays a loading spinner and the text "Processing.....". Below this, it states: "This will take a few minutes depending on the input file size. A notification email with the result link will be sent to you once the job completed." The browser's developer console on the right shows a series of JavaScript logs from job_tracking.js, including jobid, status, and tracking information, indicating a successful job completion.

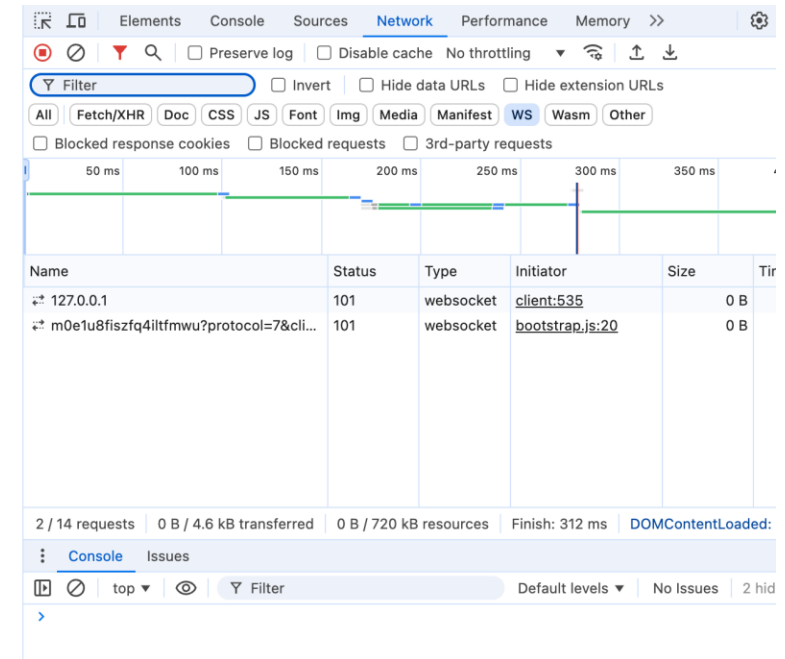
send JS request every 3 sec

Laravel – Broadcasting

WebSocket connection

```
fire a model

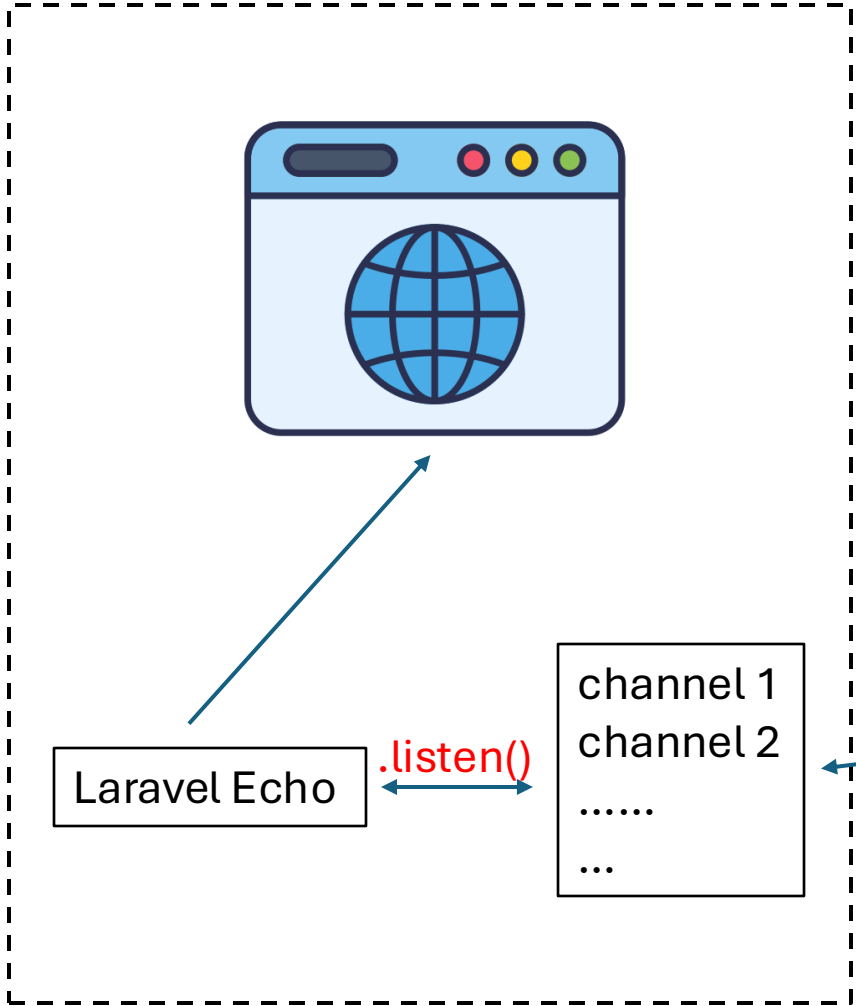
Using cpu
Building Atchley Factors dictionary
Building MHC dictionary... This might take a few seconds
Initializing TCR+CDR3 encoders
Attempt to load the VA encoder
Success
Attempt to load the VB encoder
Success
Attempt to load the CDR3A encoder
Success
Attempt to load the CDR3B encoder
Success
Initializing pMHC encoders
Attempt to load the pMHC encoder
Success
```



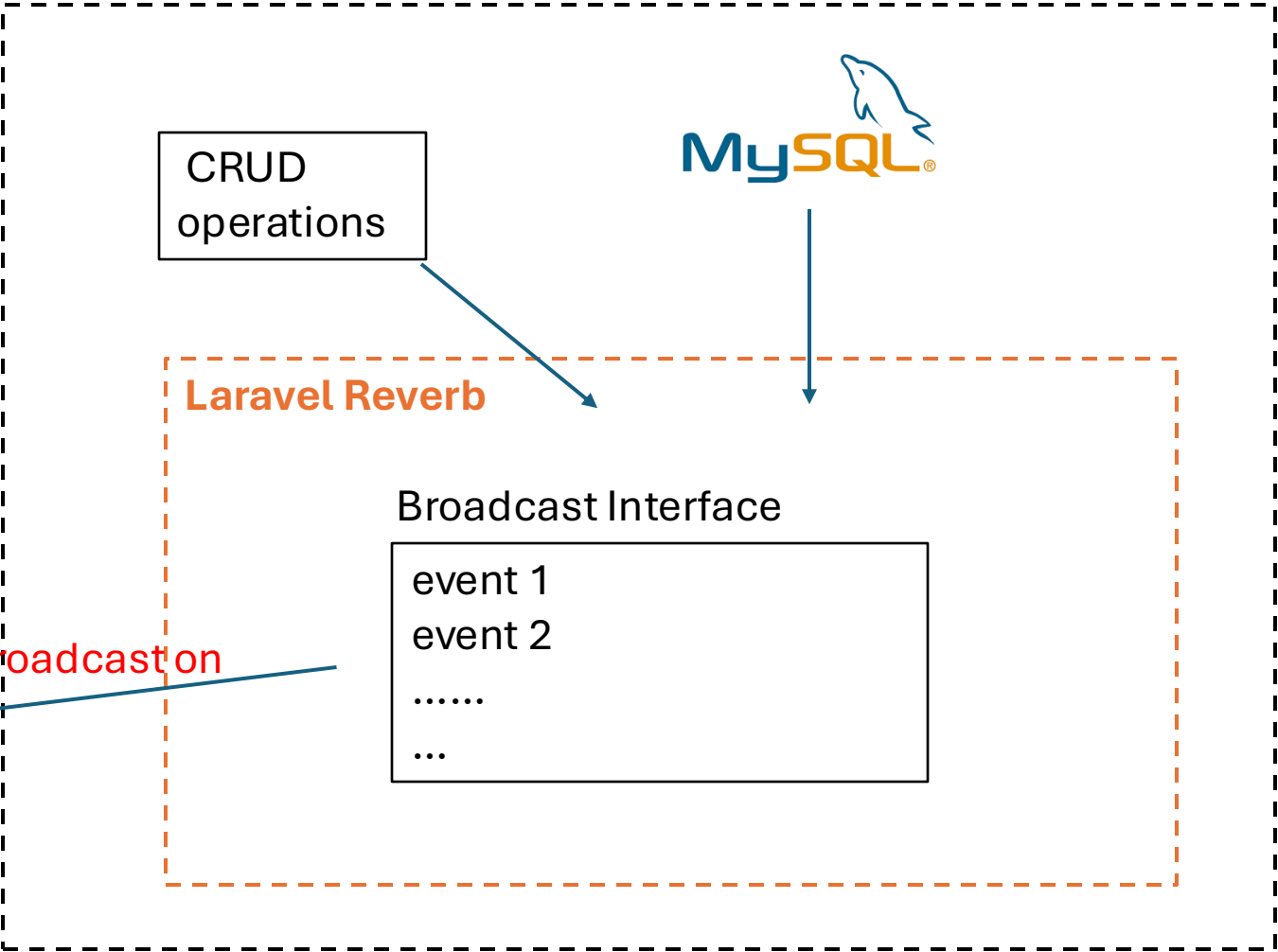
real time update with WS connection built up

Laravel – Broadcasting

Frontend



Backend



Laravel – Broadcasting

Advantages

1. better user experience
2. broader practical scenarios
 - live chat box with LLM
 - Real-time Collaborative Editing
 - Real-time data visualization

Trade-offs

1. More effort on deployment
2. Connection workload and restrictions at server side

Thanks for Listening!