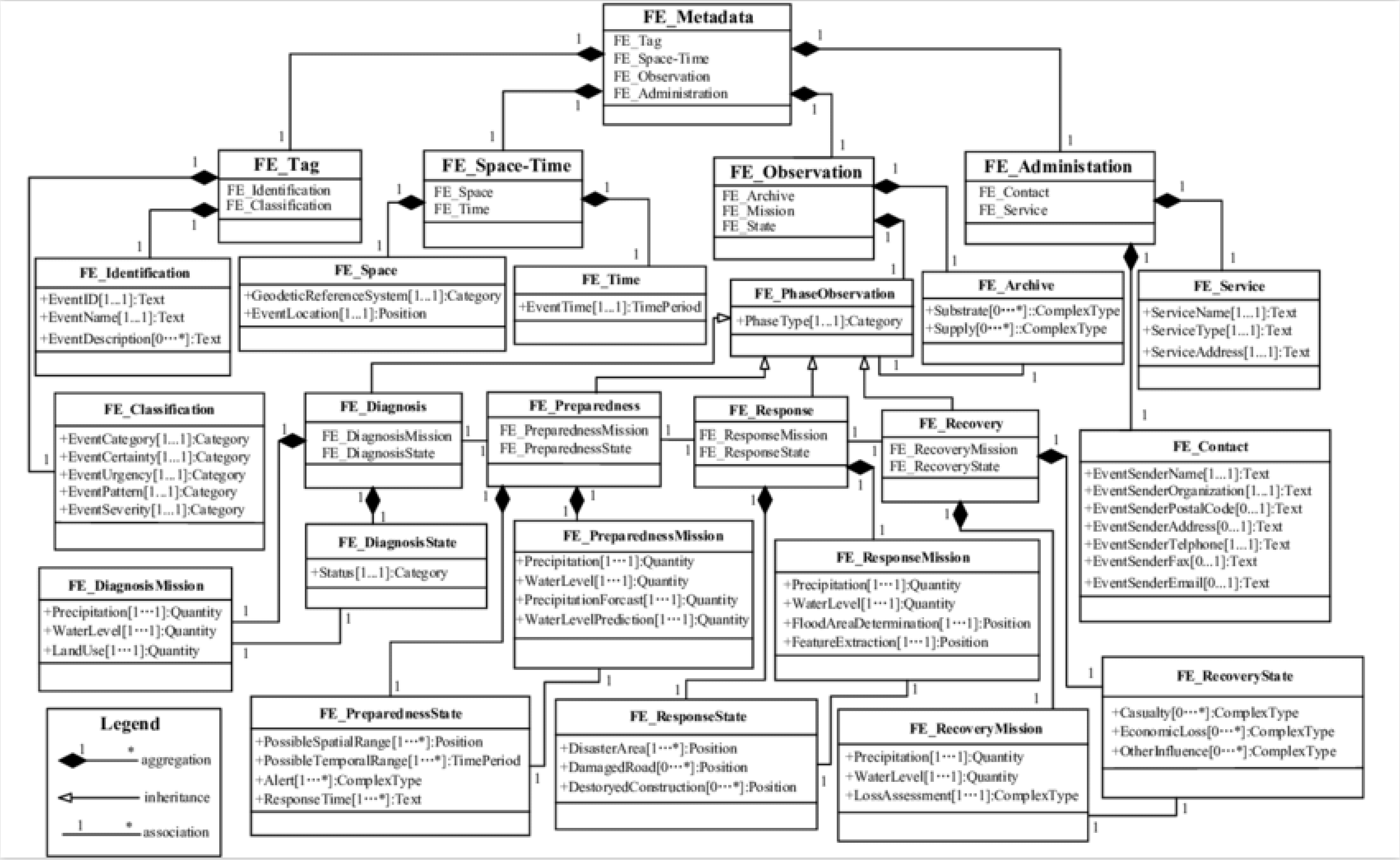
SOFTWARE MEASURES

A small-scale case study

MOTIVATION


FE_Metadata
FE_Tag
FE_Space-Time
FE_Observation
FE_Administration
FE_Tag
FE_Identification
FE_Classification
FE_Space-Time
FE_Space
FE_Time
FE_Observation
FE_Archive
FE_Mission
FE_State
FE_Administation
FE_Contact
FE_Service
FE_Identification
+EventID[1...1]:Text
+EventName[1...1]:Text
+EventDescription[0⋯*]:Text
FE_Space
+GeodeticReferenceSystem[1...1]:Category
+EventLocation[1...1]:Position
FE_Time
+EventTime[1...1]:TimePeriod
FE_PhaseObservation
+PhaseType[1...1]:Category
FE_Archive
+Substrate[0⋯*]::ComplexType
+Supply[0⋯*]::ComplexType
FE_Service
+ServiceName[1...1]:Text
+ServiceType[1...1]:Text
+ServiceAddress[1...1]:Text
FE_Classification
+EventCategory[1...1]:Category
+EventCertainty[1...1]:Category
+EventUrgency[1...1]:Category
+EventPattern[1...1]:Category
+EventSeverity[1...1]:Category
FE_Diagnosis
FE_DiagnosisMission
FE_DiagnosisState
FE_Preparedness
FE_PreparednessMission
FE_PreparednessState
FE_Response
FE_ResponseMission
FE_ResponseState
FE_Recovery
FE_RecoveryMission
FE_RecoveryState
FE_Contact
+EventSenderName[1...1]:Text
+EventSenderOrganization[1...1]:Text
+EventSenderPostalCode[0...1]:Text
+EventSenderAddress[0...1]:Text
+EventSenderTelphone[1...1]:Text
+EventSenderFax[0...1]:Text
+EventSenderEmail[0...1]:Text
FE_DiagnosisState
+Status[1...1]:Category
FE_PreparednessMission
+Precipitation[1⋯1]:Quantity
+WaterLevel[1⋯1]:Quantity
+PrecipitationForcast[1⋯1]:Quantity
+WaterLevelPrediction[1⋯1]:Quantity
FE_ResponseMission
+Precipitation[1⋯1]:Quantity
+WaterLevel[1⋯1]:Quantity
+FloodAreaDetermination[1⋯1]:Position
+FeatureExtraction[1⋯1]:Position
FE_DiagnosisMission
+Precipitation[1⋯1]:Quantity
+WaterLevel[1⋯1]:Quantity
+LandUse[1⋯1]:Quantity
FE_RecoveryState
+Casualty[0⋯*]:ComplexType
+EconomicLoss[0⋯*]:ComplexType
+OtherInfluence[0⋯*]:ComplexType
Legend
aggregation
inheritance
association
FE_PreparednessState
+PossibleSpatialRange[1⋯*]:Position
+PossibleTemporalRange[1⋯*]:TimePeriod
+Alert[1⋯*]:ComplexType
+ResponseTime[1⋯*]:Text
FE_ResponseState
+DisasterArea[1⋯*]:Position
+DamagedRoad[0⋯*]:Position
+DestoryedConstruction[0⋯*]:Position
FE_RecoveryMission
+Precipitation[1⋯1]:Quantity
+WaterLevel[1⋯1]:Quantity
+LossAssessment[1⋯1]:ComplexType

MOTIVATION

Complex Software + Principles of Good Programming
Software Architectures
Frameworks = Problem solved?

Subjective:
perceived **maintainability**, perceived **complexity**, perceived **reusability**…

Objective: (through Software Measurement)
Maintainability Index – Score, Cyclomatic Complexity, Coupling Measures…

APPLICATIONS OF SOFTWARE MEASUREMENTS

- Cost Estimation
  - Function Point Measurement
  - COCOMO II
- Productivity Measures
  - Halstead Measures
  - Maintainability Index
- Complexity Measures
  - McCabes Cyclomatic Complexity
  - Halstead Measures
  - Weighted Method Count
- Quality Models
  - DeLone and McLean Information System Success Model
  - McCall
  - Boehm

Thesis

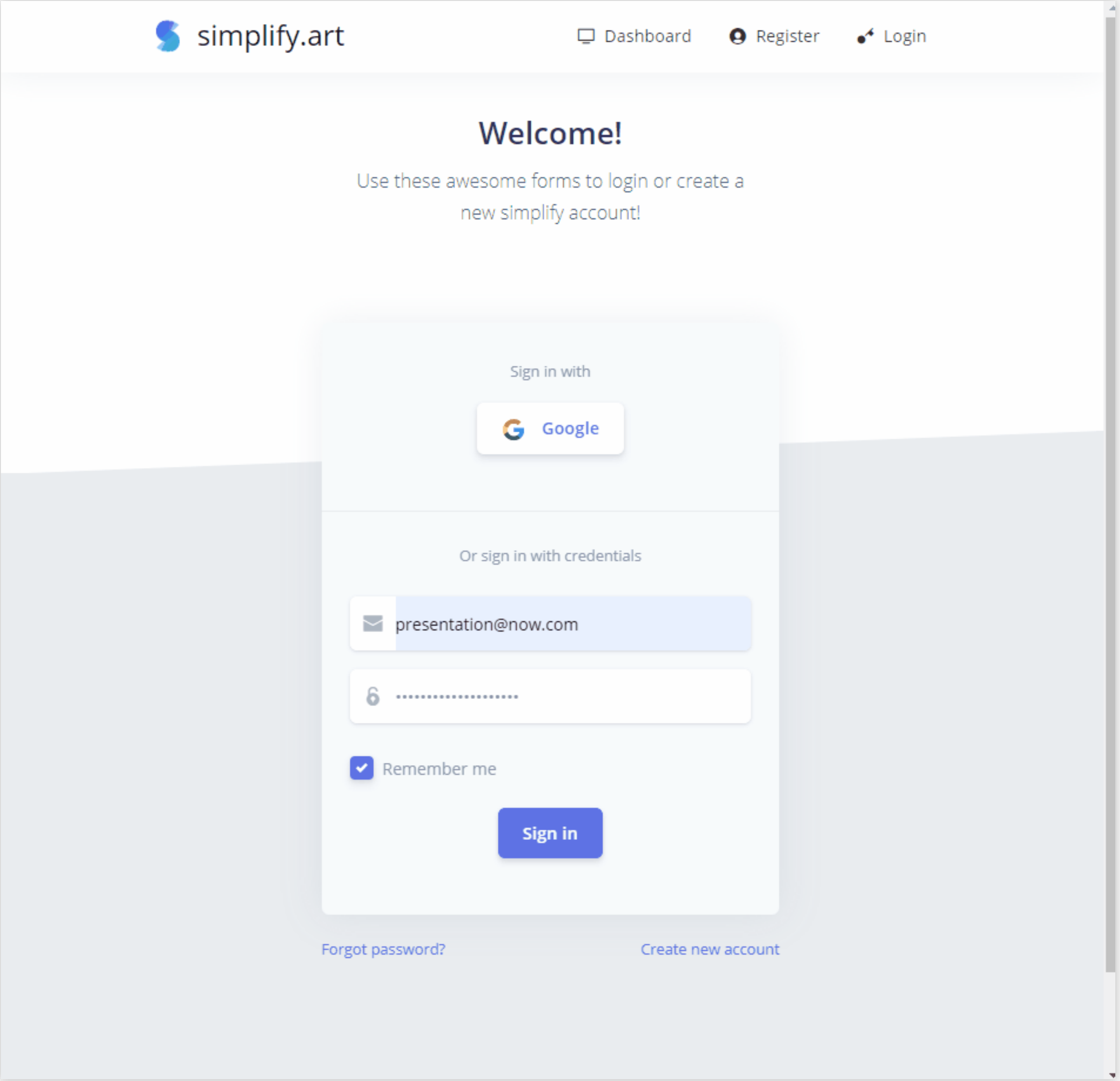
GOALS

- Use a set of **quantitative software measurements** to compare two web frameworks
- Investigate **evolution** of **software attributes**
  - Find a **set** of **software measurements**
  - Check if set is **applicable to chosen frameworks** (ReactJS and Laravel)
  - Check if certain **attributes of the architecture** have **expected impact**
  - Check if set is **applicable** for **framework comparison**

EXPERIMENT

- Choose a **set of software measurements**
- Find **tools to calculate** these measurements
- Develop the **same application twice**
  - ReactJS
  - Laravel
- Conduct measurements after **addition of specific features**
- Evaluation of results

APPLICATION



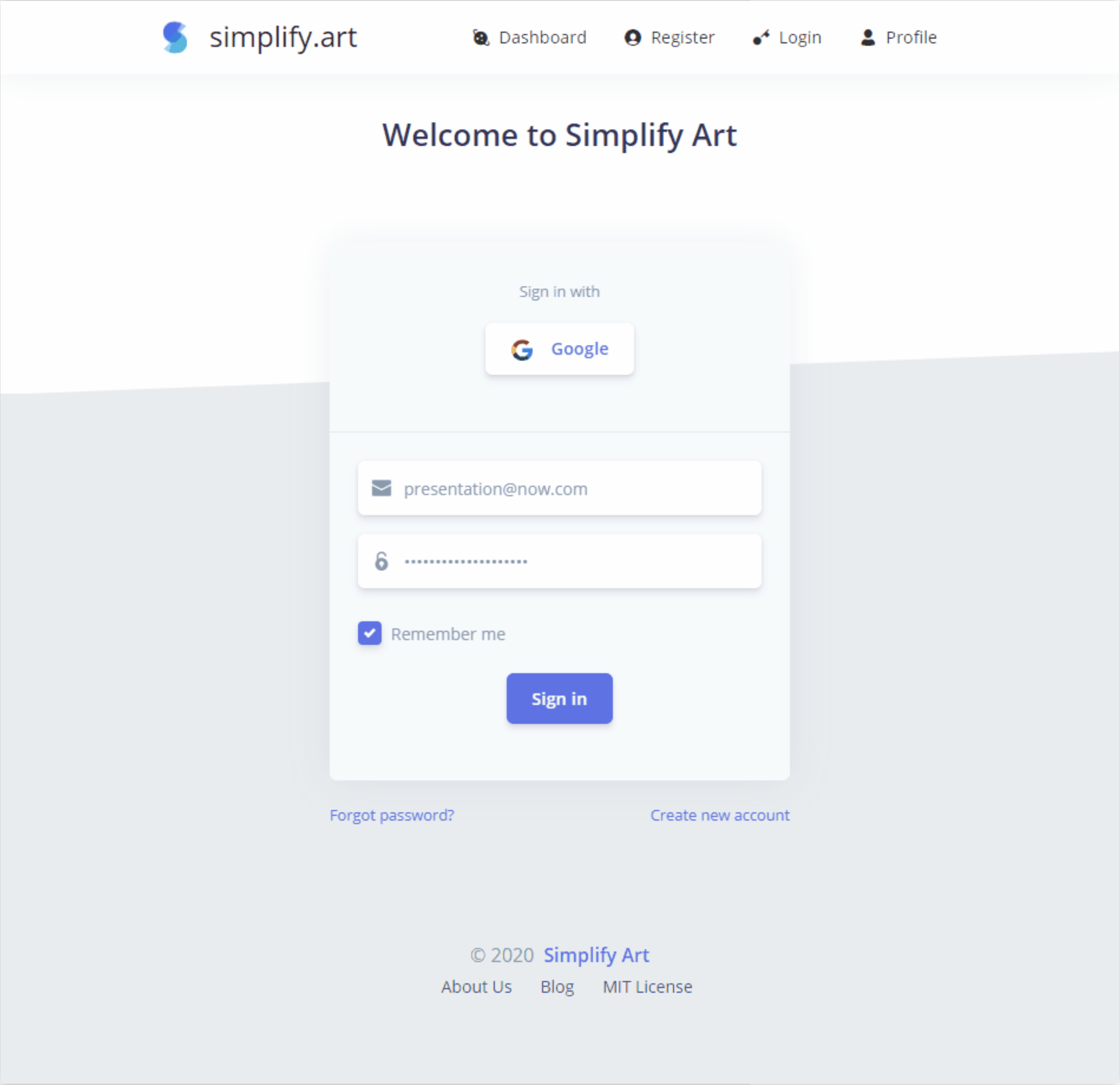
Welcome!

Use these awesome forms to login or create a new simplify account!

presentation@now.com

••••••••••••••••••••

☑ Remember me



Welcome to Simplify Art

presentation@now.com

••••••••••••••••••••

☑ Remember me

© 2020 Simplify Art

Measurements and Tools

MEASUREMENTS

- **Complexity**
  1. McCabes Cyclomatic Complexity
  2. Halstead Measures

- Quality Attributes
  - **Maintainability**
    1. Maintainability Index
  - **Reusability + Flexibility**
    - Coupling Measures
      1. Afferent-Coupling
      2. Efferent-Coupling
      3. Instability

TOOLS

ReactJS

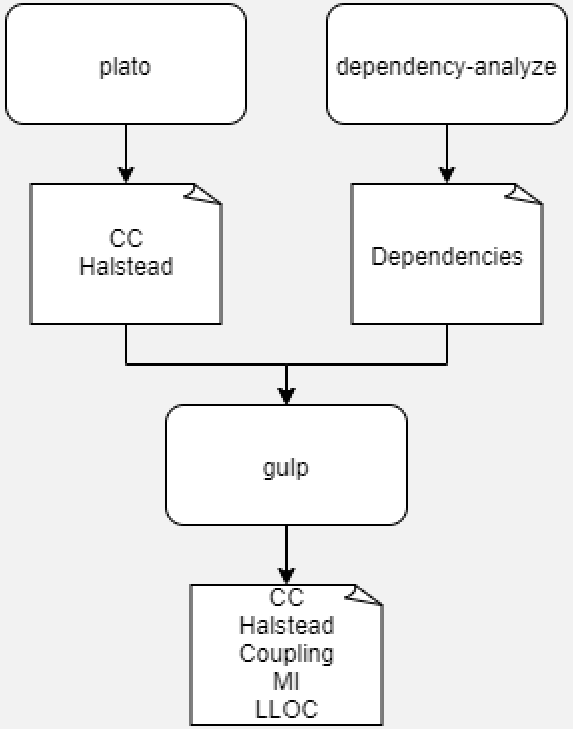
- plato
  - Cyclomatic Complexity
  - Halstead Measures
  - LLOC*
  - Maintainability Index*
- dependency-analyze
  - Coupling

Laravel

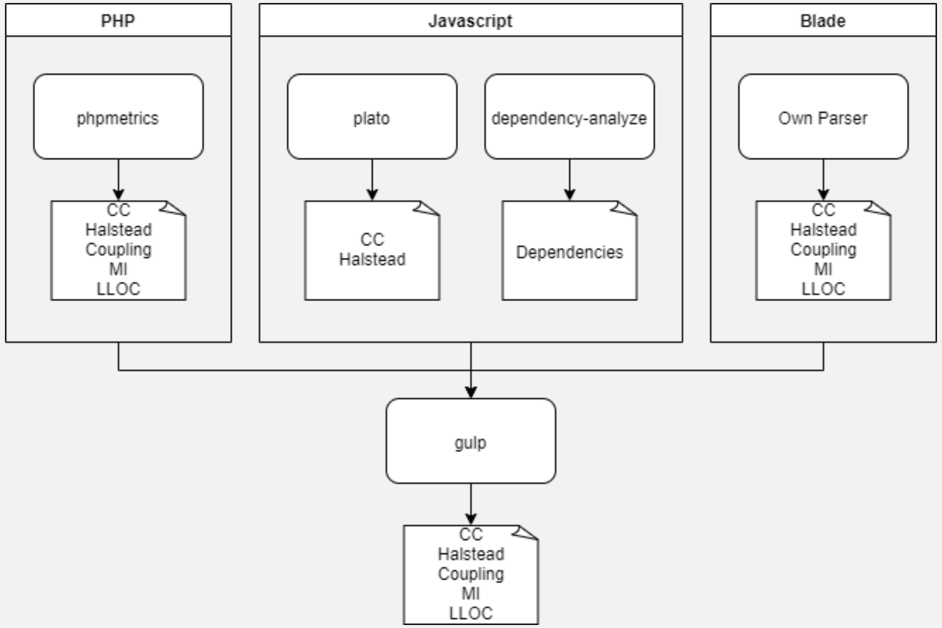
- phpmetrics
  - Cyclomatic Complexity
  - Halstead Measures
  - LLOC
  - Maintainability Index
  - Coupling
- plato (JS only)
- own parser (Blade only)

TOOLS – PIPELINE - OVERVIEW

ReactJS

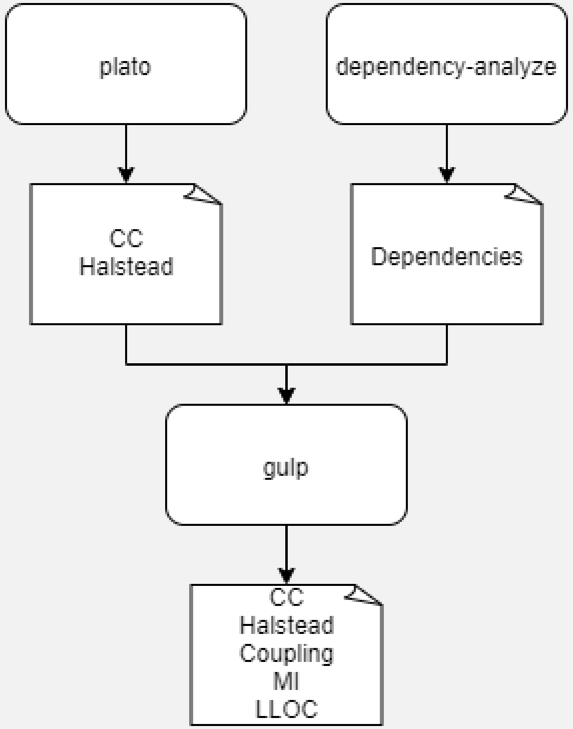


Laravel



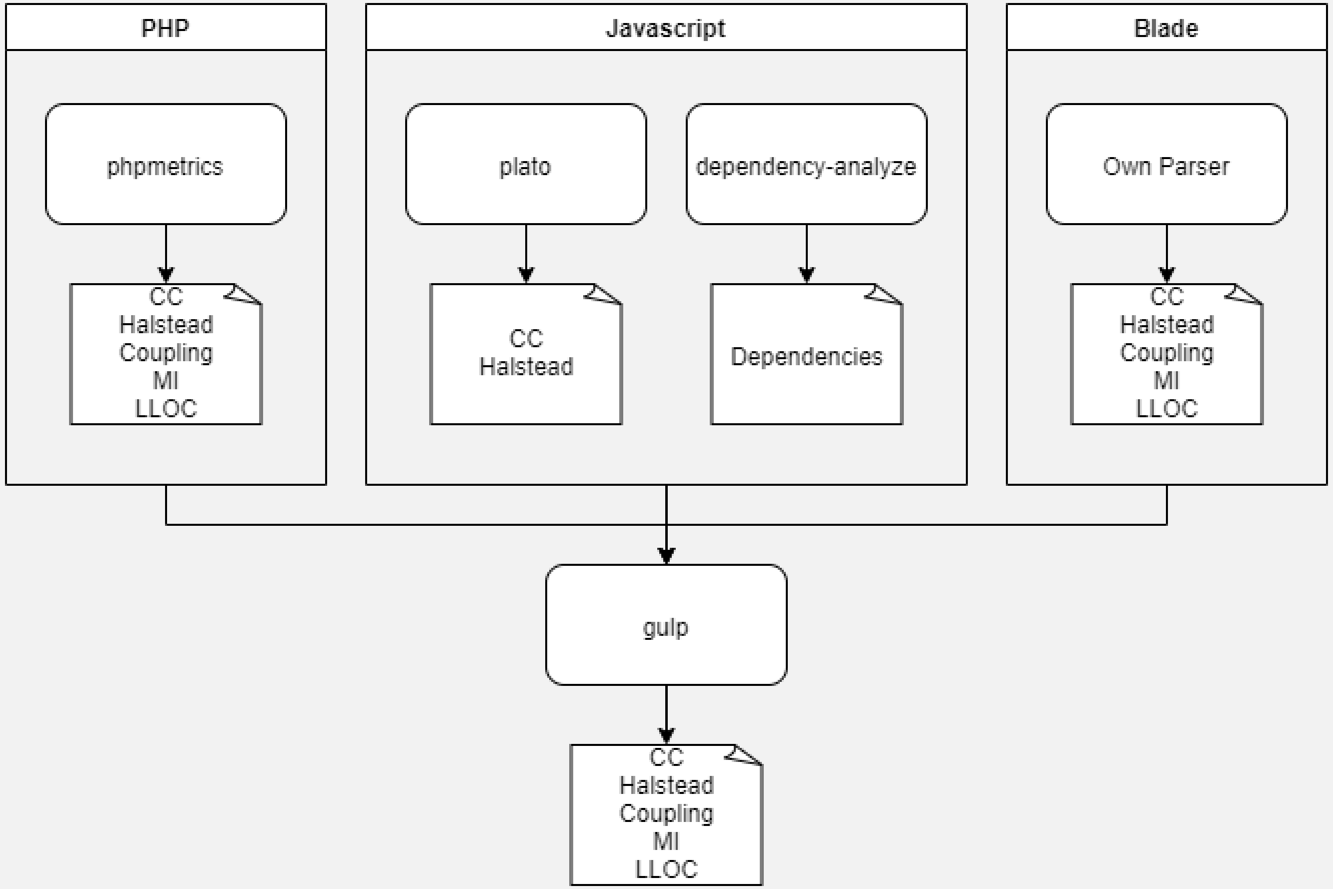
TOOLS – PIPLINE

ReactJS



TOOLS – PIPELINE

Laravel



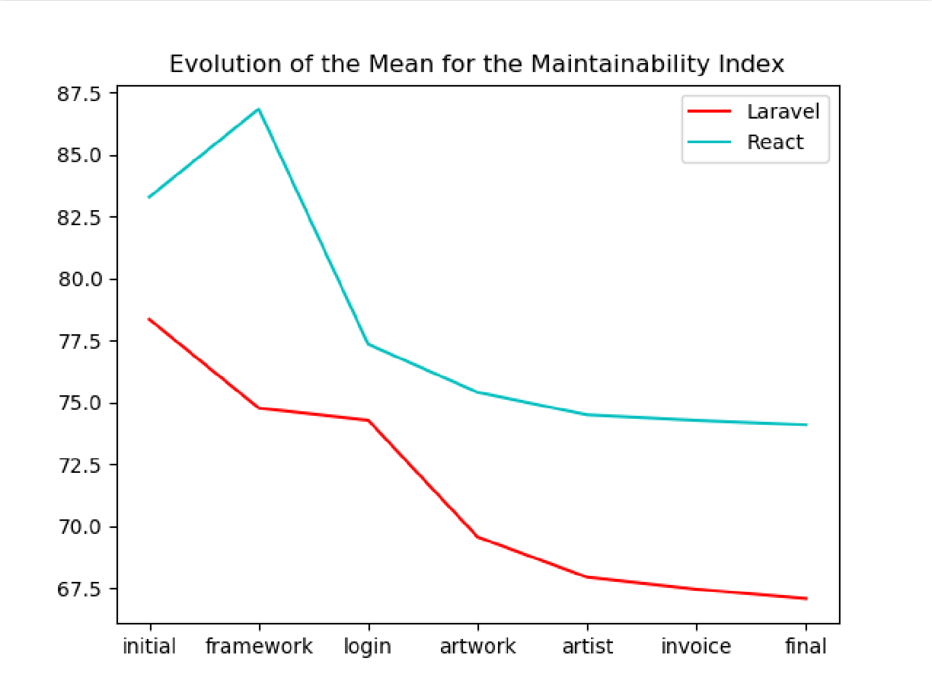
Pitfalls

PITFALLS – CONSISTENCY

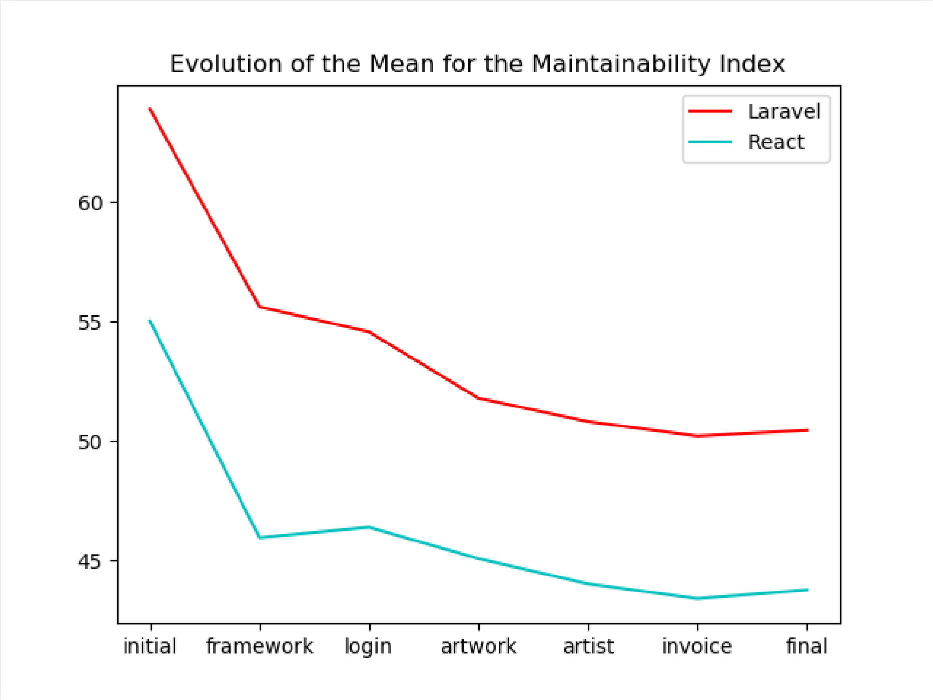
- Different tools **calculate** software measurements **differently**
  - Definition of **how to calculate** a given software measure
  - Definition of **parts** needed for calculation

PITFALLS – CONSISTENCY

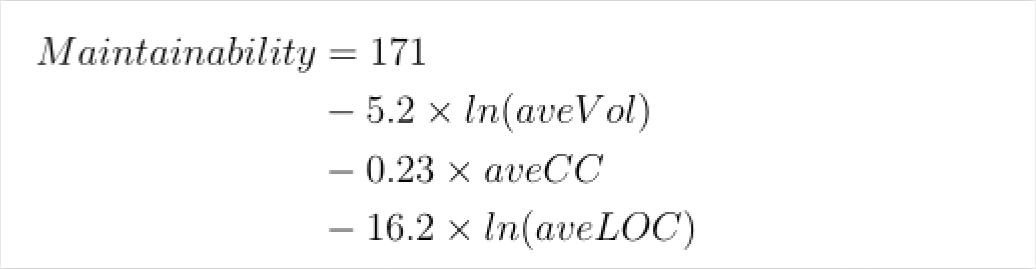
Before



After

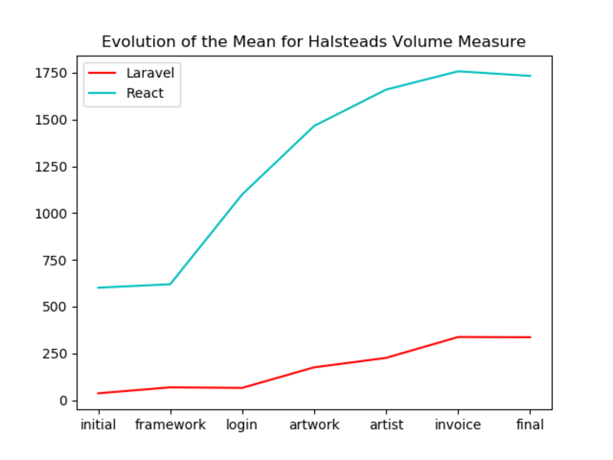


MAINTAINABILITY INDEX

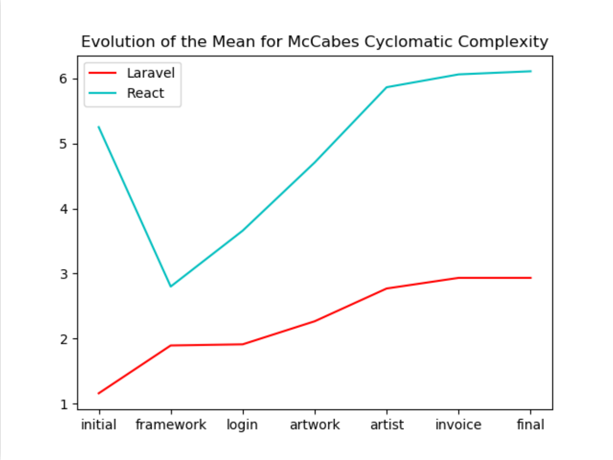


$$\begin{aligned} Maintainability = 171 &\\ - 5.2 \times ln(aveVol) &\\ - 0.23 \times aveCC &\\ - 16.2 \times ln(aveLOC) & \end{aligned}$$

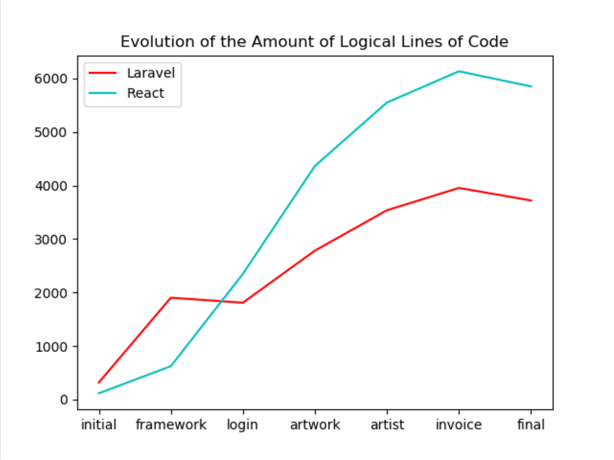
PITFALLS – CONSISTENCY



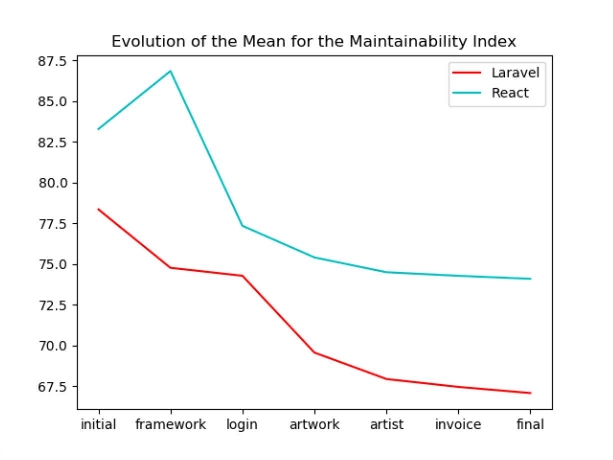
Evolution of the Mean for Halsteads Volume Measure



Evolution of the Mean for McCabes Cyclomatic Complexity

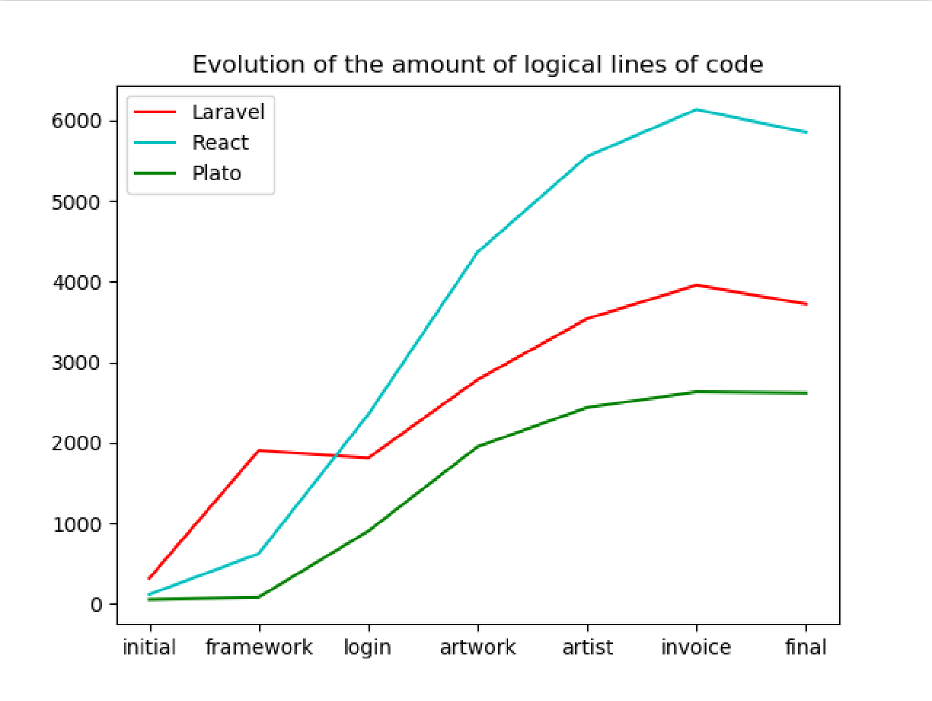


Evolution of the Amount of Logical Lines of Code



Evolution of the Mean for the Maintainability Index

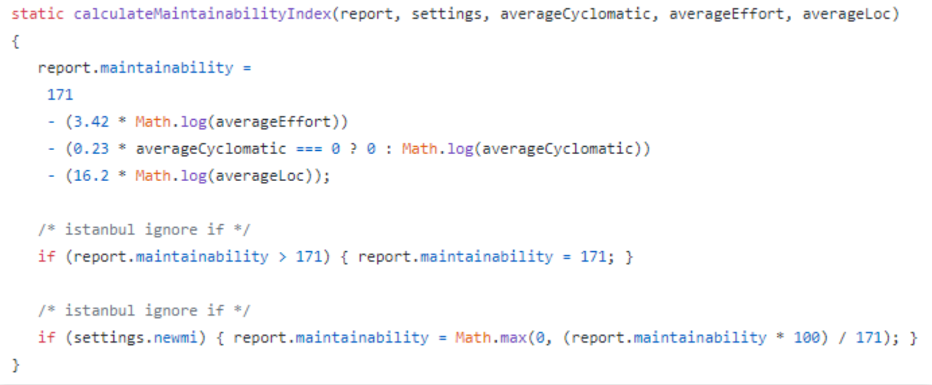
PITFALLS – CONSISTENCY



- phpmetrics & own ReactJS tool
  - Remove all **comments**
  - Remove **empty lines**
  - Remove lines which contain **only curly braces**
- plato
  - Generate parse tree
  - Every statement = LLOC

MAINTAINABILITY INDEX – CALCULATION

plato

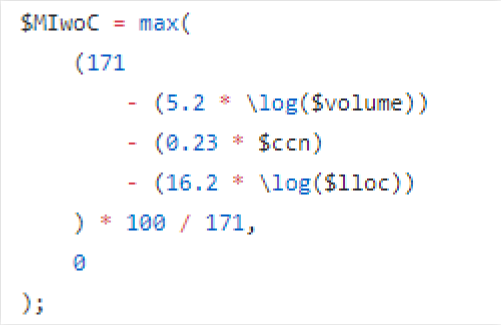


```
static calculateMaintainabilityIndex(report, settings, averageCyclomatic, averageEffort, averageLoc)
{
   report.maintainability =
    171
    - (3.42 * Math.log(averageEffort))
    - (0.23 * averageCyclomatic === 0 ? 0 : Math.log(averageCyclomatic))
    - (16.2 * Math.log(averageLoc));

   /* istanbul ignore if */
   if (report.maintainability > 171) { report.maintainability = 171; }

   /* istanbul ignore if */
   if (settings.newmi) { report.maintainability = Math.max(0, (report.maintainability * 100) / 171); }
}
```

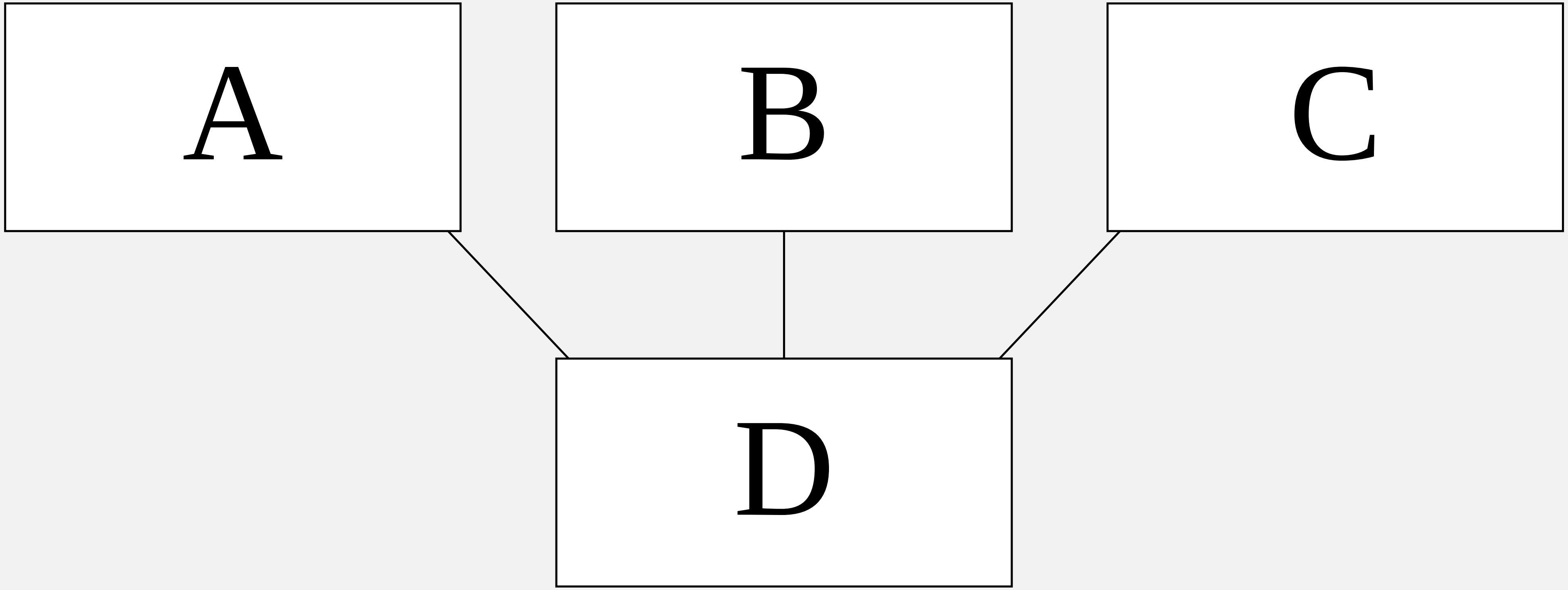
phpmetrics



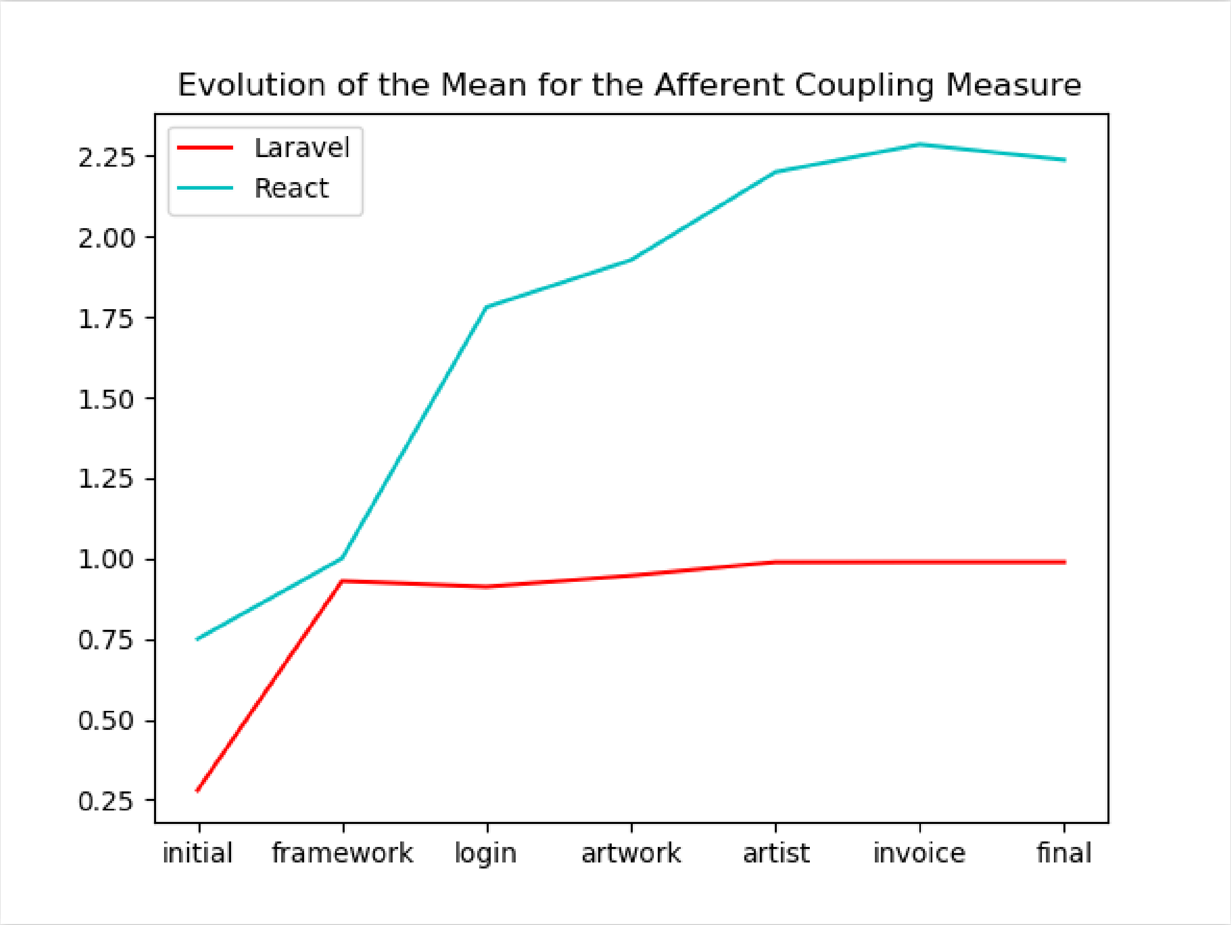
```
$MIwoC = max(
    (171
        - (5.2 * \log($volume))
        - (0.23 * $ccn)
        - (16.2 * \log($lloc))
    ) * 100 / 171,
    0
);
```

Results

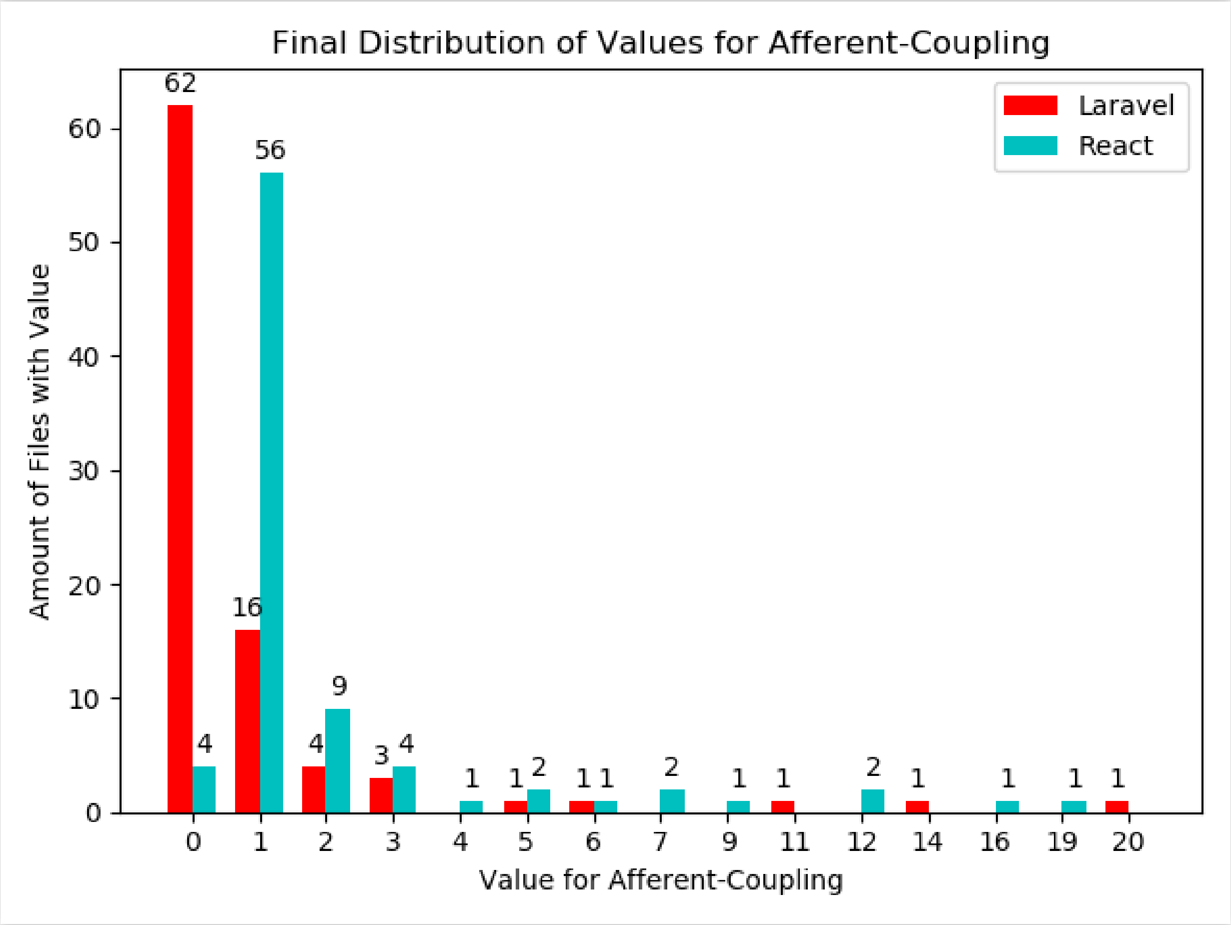
AFFERENT - COUPLING



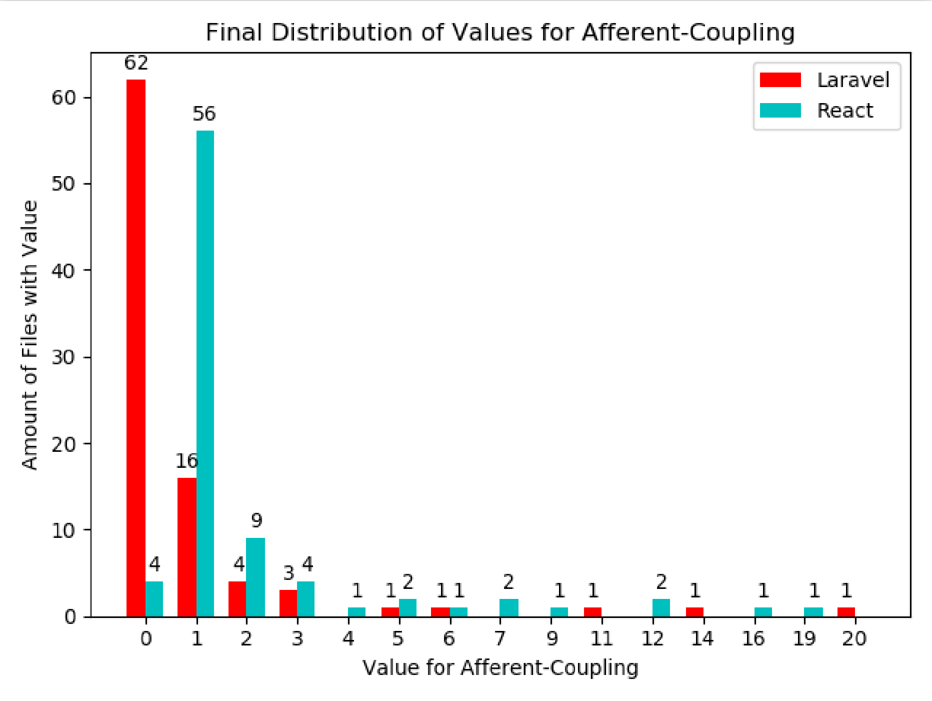
AFFERENT - COUPLING



AFFERENT - COUPLING

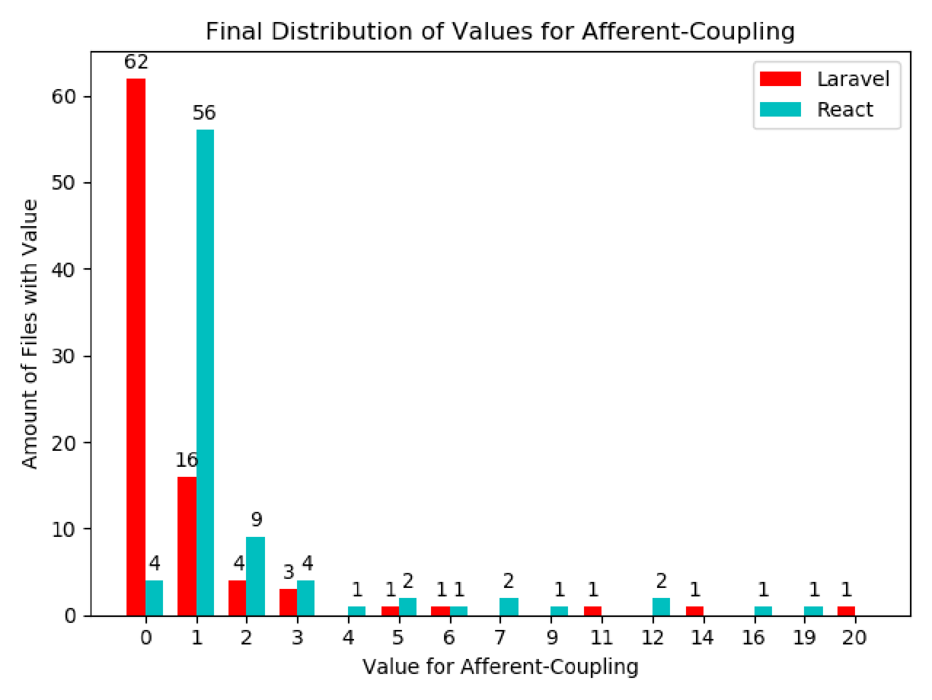


AFFERENT - COUPLING



- High Afferent-Coupling
  - Difficult to change
- Afferent-Coupling = 0
  - Entry files or unused files

AFFERENT - COUPLING



- **Not applicable** to the Laravel framework
    - Import statements between key components are **hidden from measurement**
    - Helper classes are called automagically

AFFERENT - COUPLING

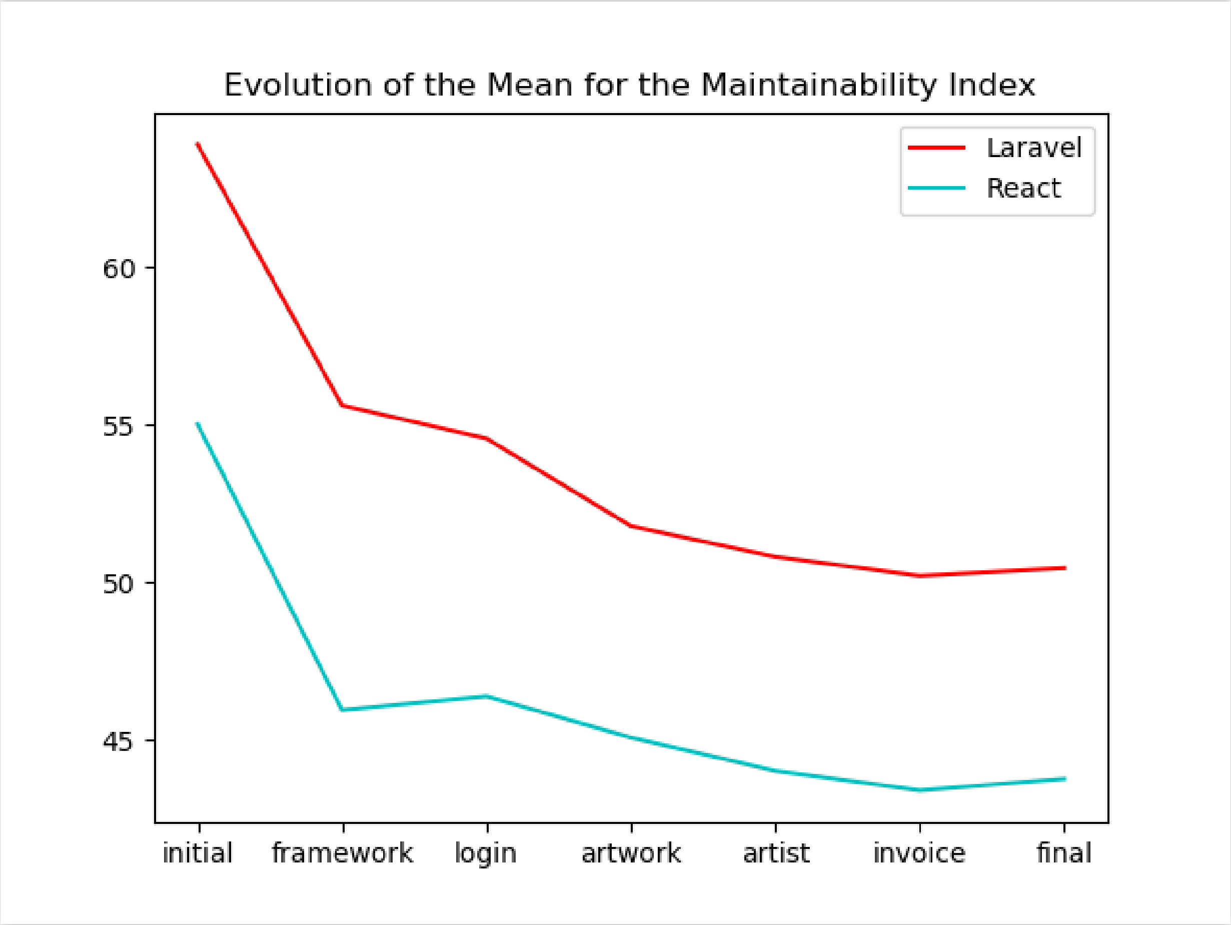
Pros

- Detect core components of the architecture
- Detect possibly rigid files
  - Split responsibility amongst multiple files to improve flexibility
- Detect reusable files
- Detect unused files

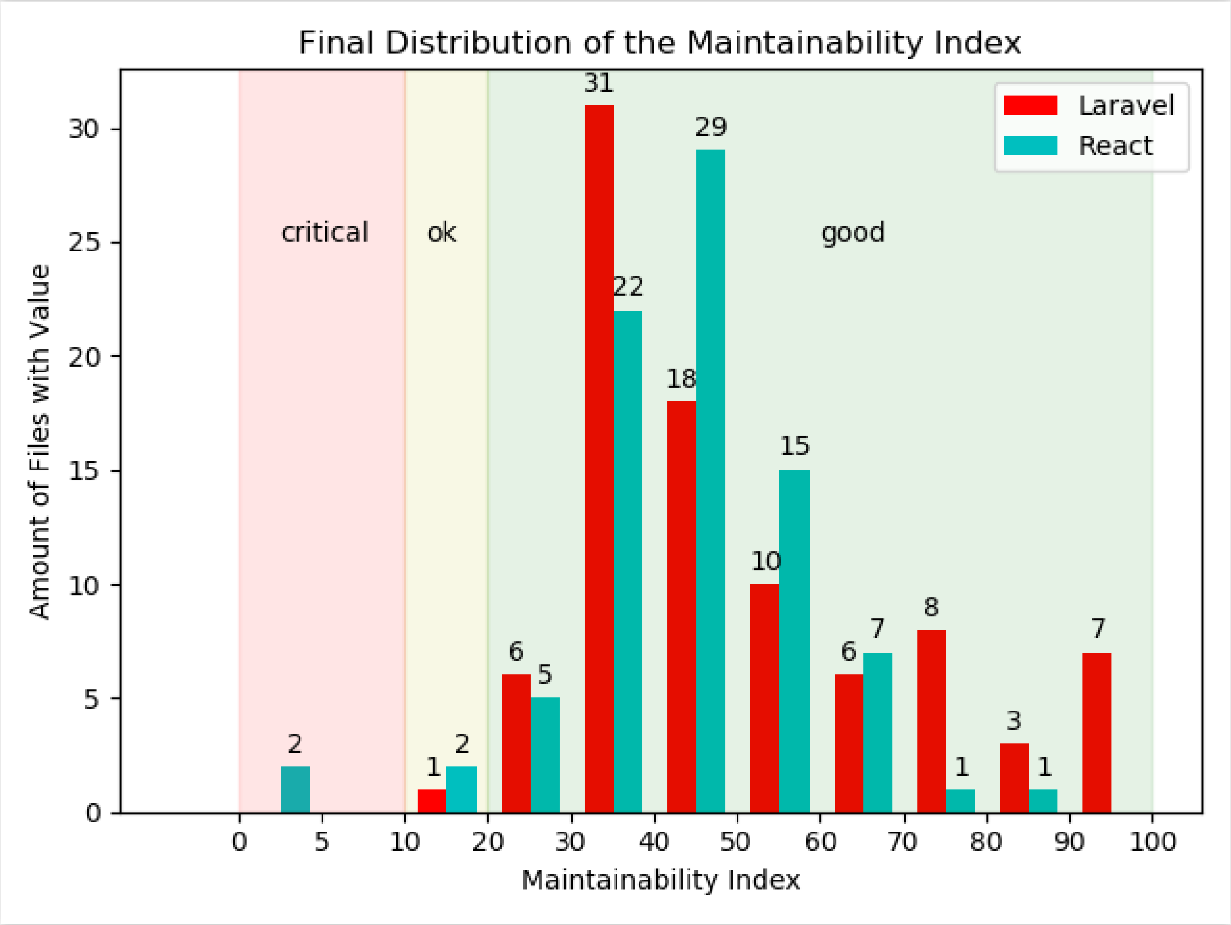
Cons

- Depends on programming language used
  - Not applicable to the Laravel framework

MAINTAINABILITY INDEX



MAINTAINABILITY INDEX



MAINTAINABILITY INDEX

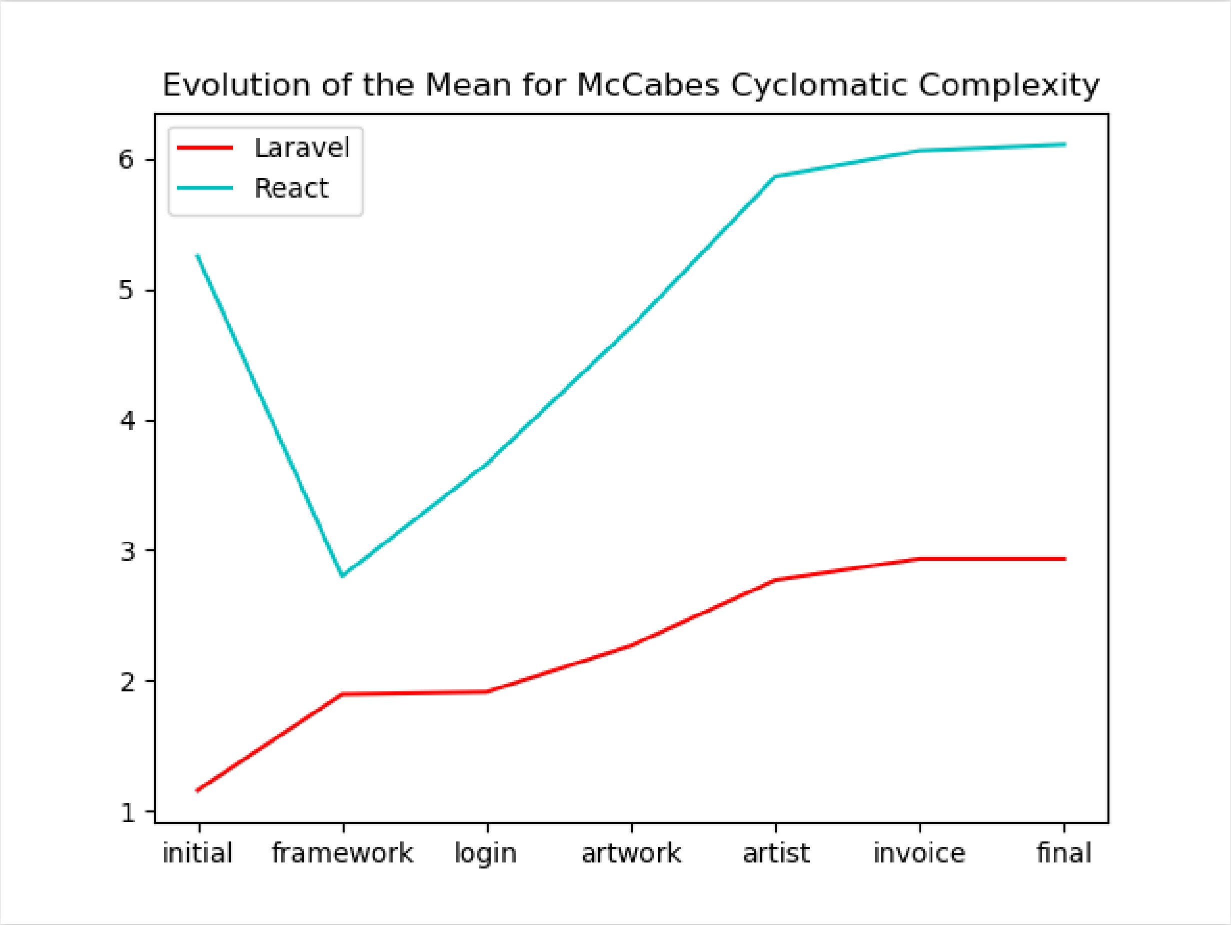
Pros

- May allow to **detect** files which are **hard to maintain**
- Seems to capture the expected changes in maintainability

Cons

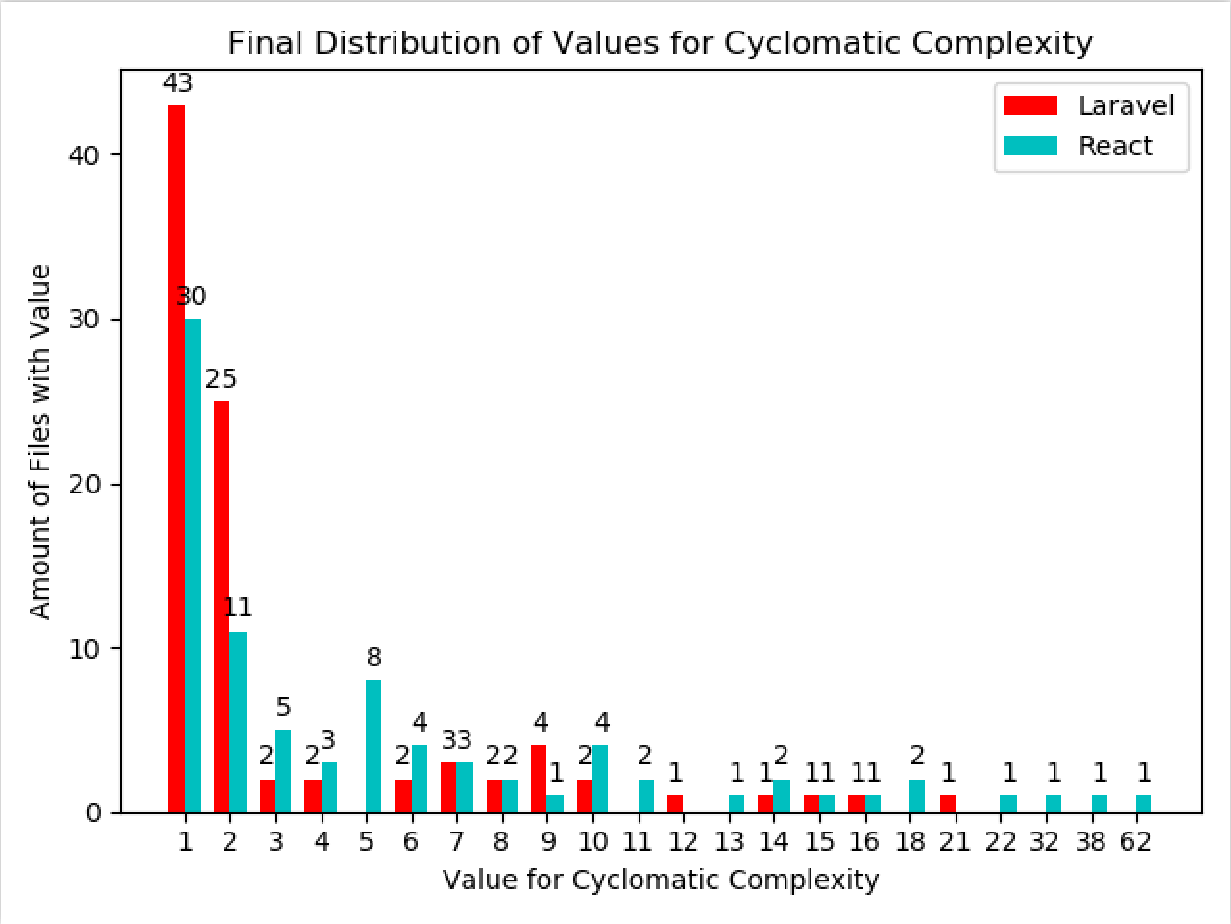
- **Unclear mapping** between resulting values and quantitative code attributes
- Mainly influenced by LLOC

CYCLOMATIC COMPLEXITY



Haslinger
Kevin

CYCLOMATIC COMPLEXITY

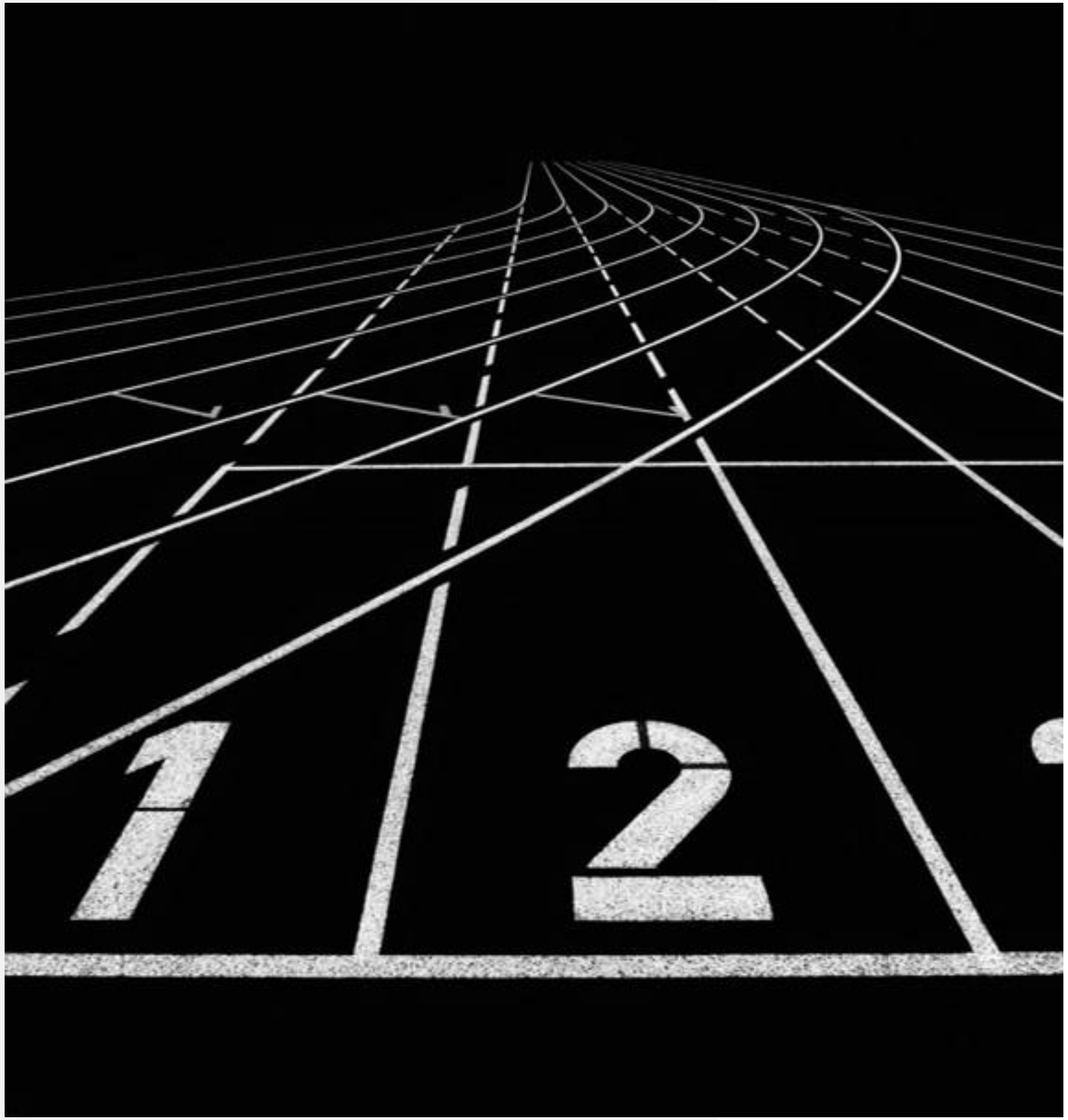


CYCLOMATIC COMPLEXITY

Pros	Cons
• Detect relatively complex files • Indicates error proneness	• Definition of complexity

“Any particular scale, sensory or physical, may be objected on the grounds of bias, low precision, restricted generality, and other factors, but the objector should remember that these are relative and practical matters and that no scale used by mortals is perfectly free of their taint.“

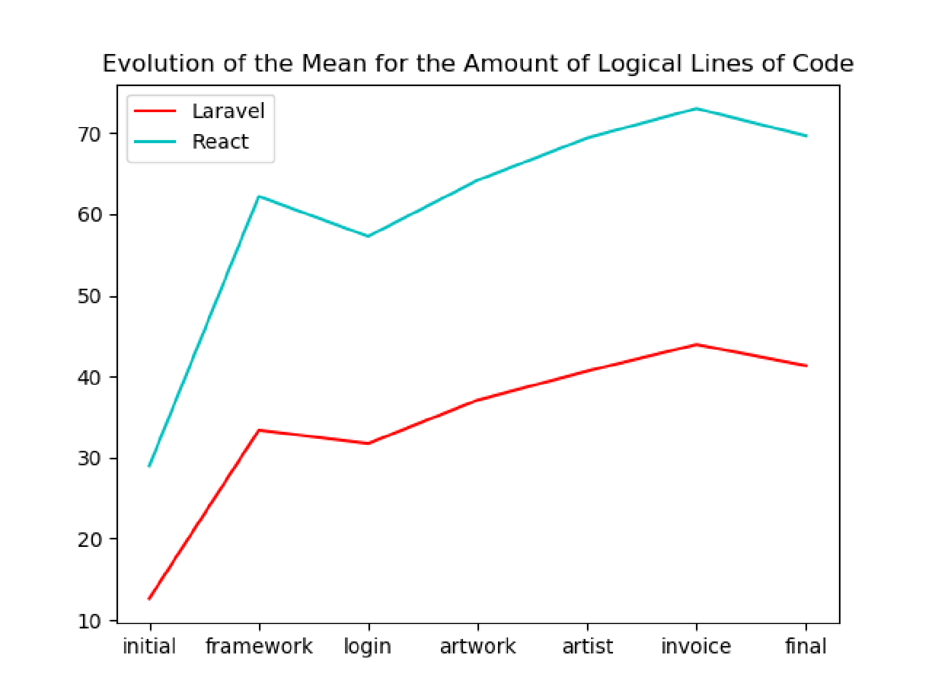
- S. S. Stevens et al. On the theory of scales of measurement. 1946.

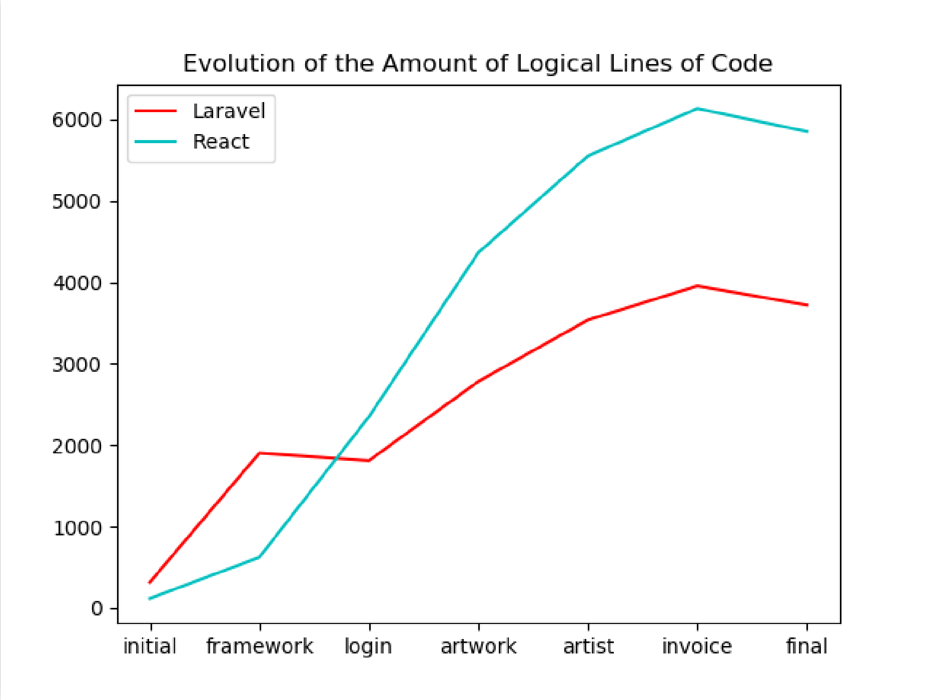


THANK YOU

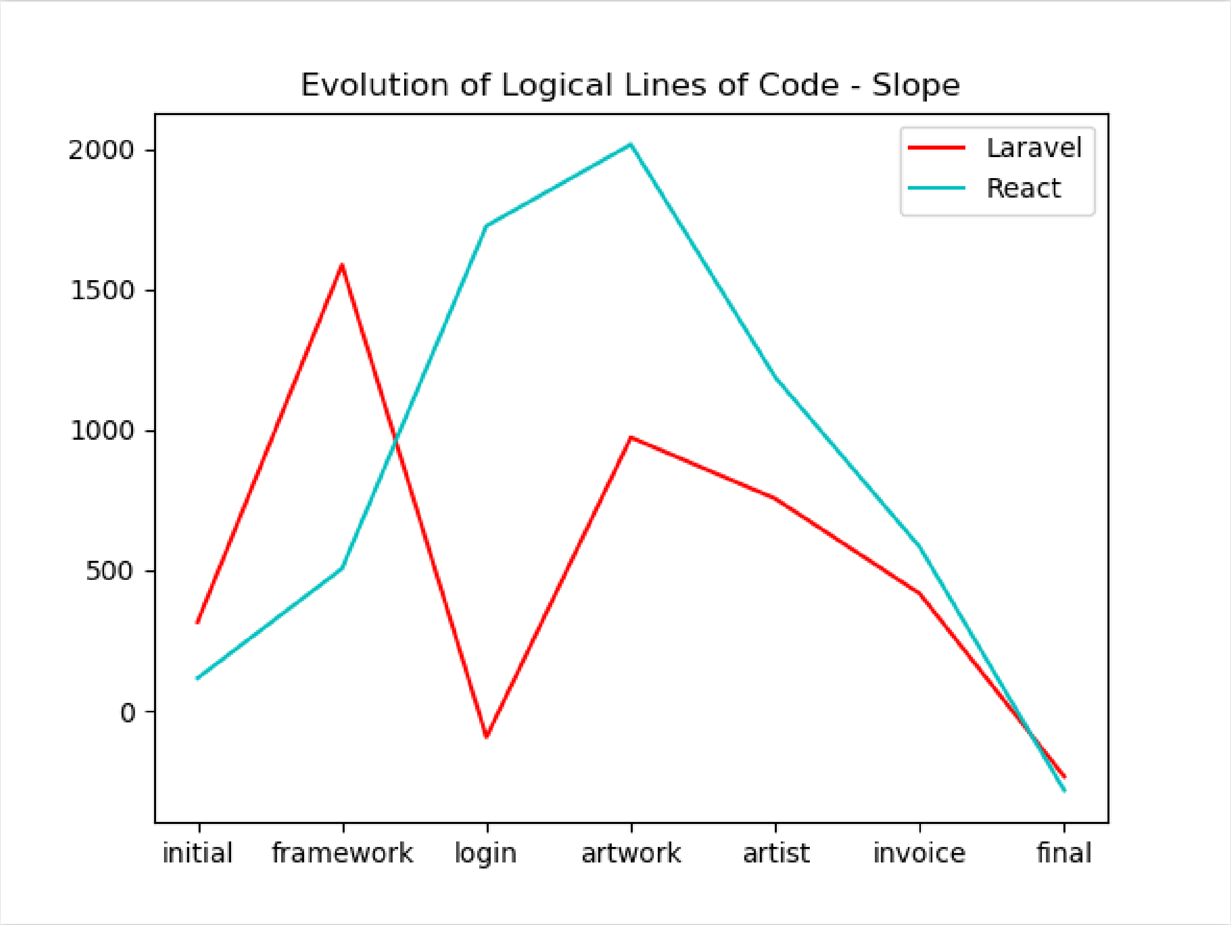
Haslinger Kevin

LOGICAL LINES OF CODE





LOGICAL LINES OF CODE



LOGICAL LINES OF CODE

Pros

- Captures changes as expected
- Captures unique attributes of frameworks
- Easy to calculate
- Clear mapping between measurement and quantitative code attributes

Cons

- Solely a measurement of size
- Depends on programming language used

GENERAL RESULTS

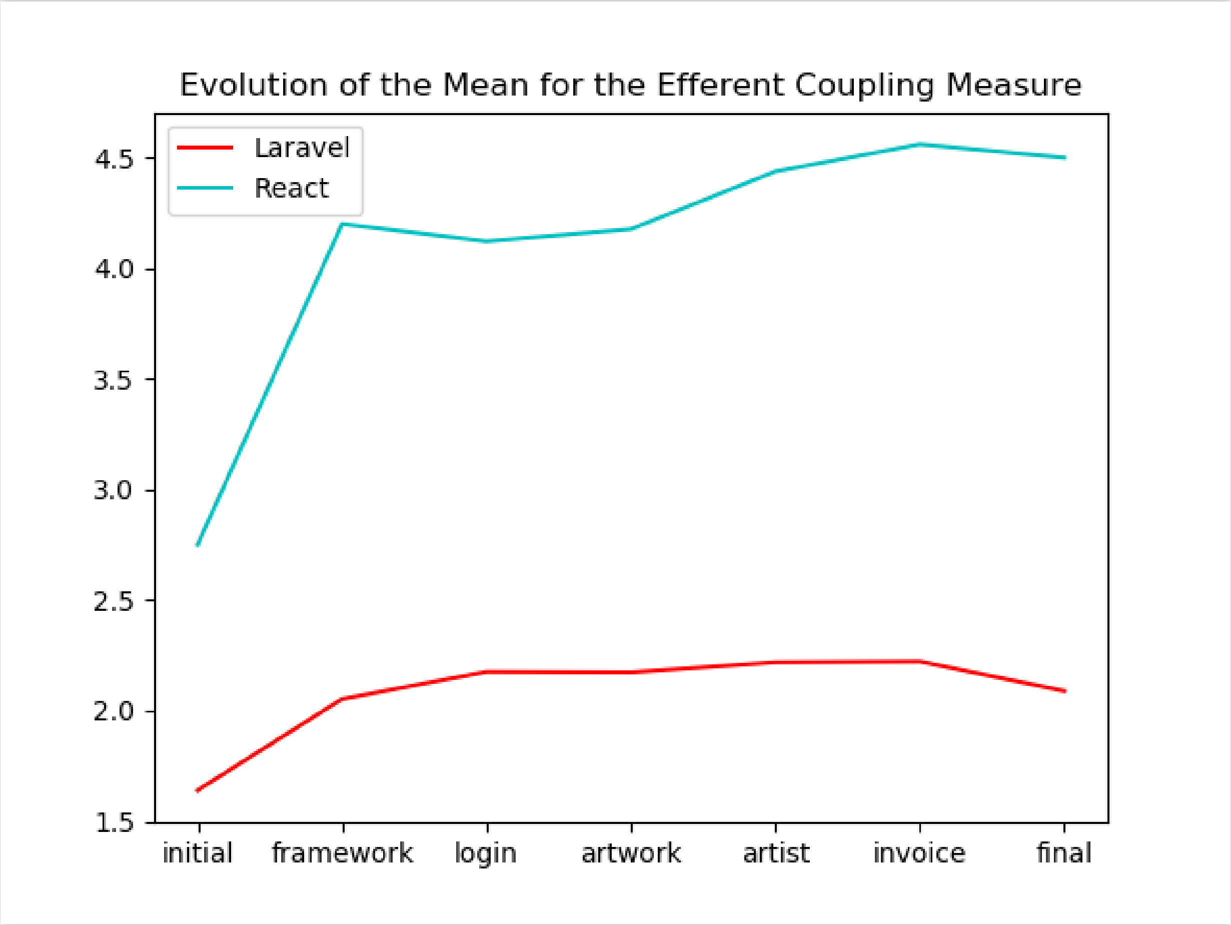
ReactJS

Measure	Applicable	Meaningful
Afferent-C	✓	✓
Efferent-C	✓	✓
Instability	✓	✓
LLOC	✓	✓
CC	✓	✓
Halstead	✓	✗
MI	✓	✗

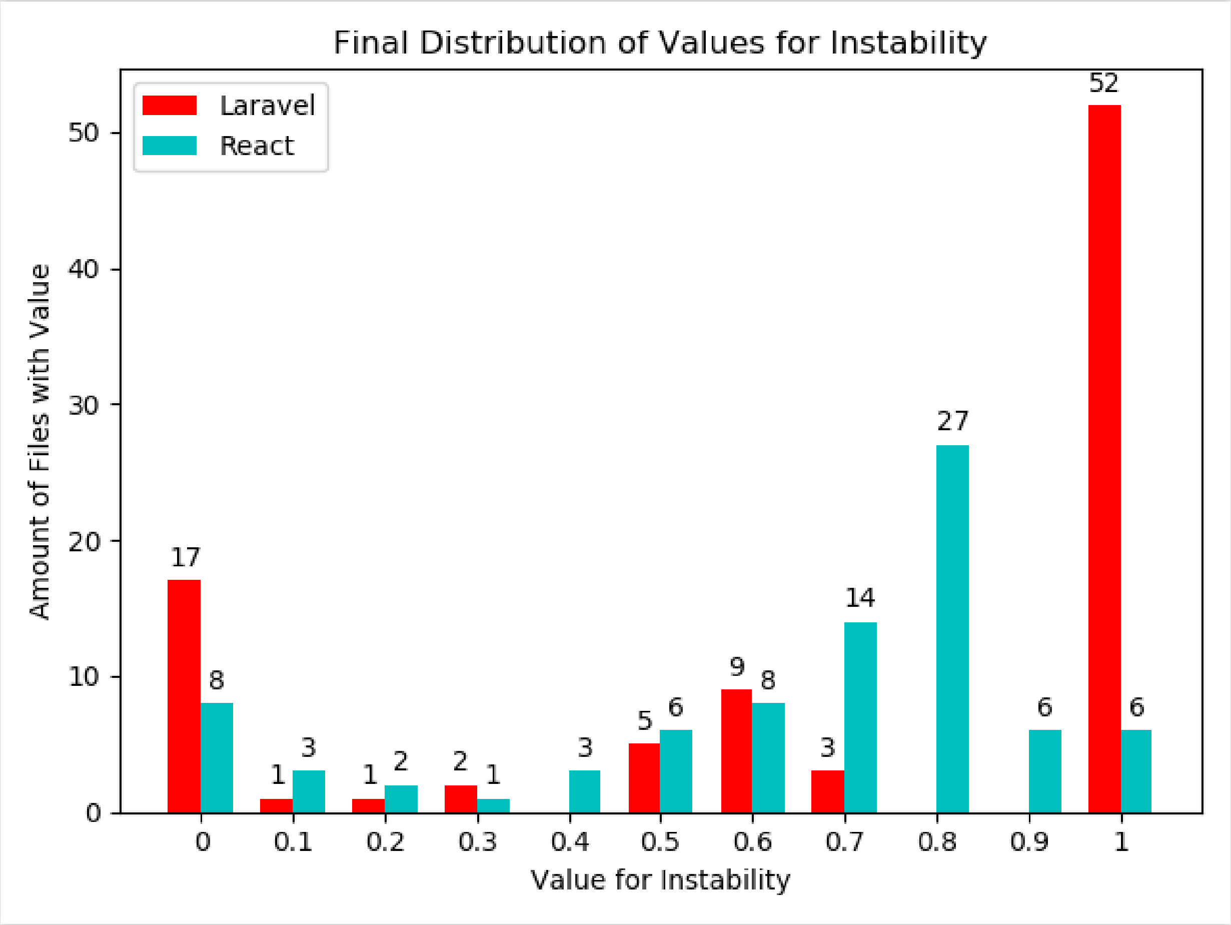
Laravel

Measure	Applicable	Meaningful
Afferent-C	✓	✗
Efferent-C	✓	✗
Instability	✓	✗
LLOC	✓	✓
CC	✓	✓
Halstead	✓	✗
MI	✓	✗

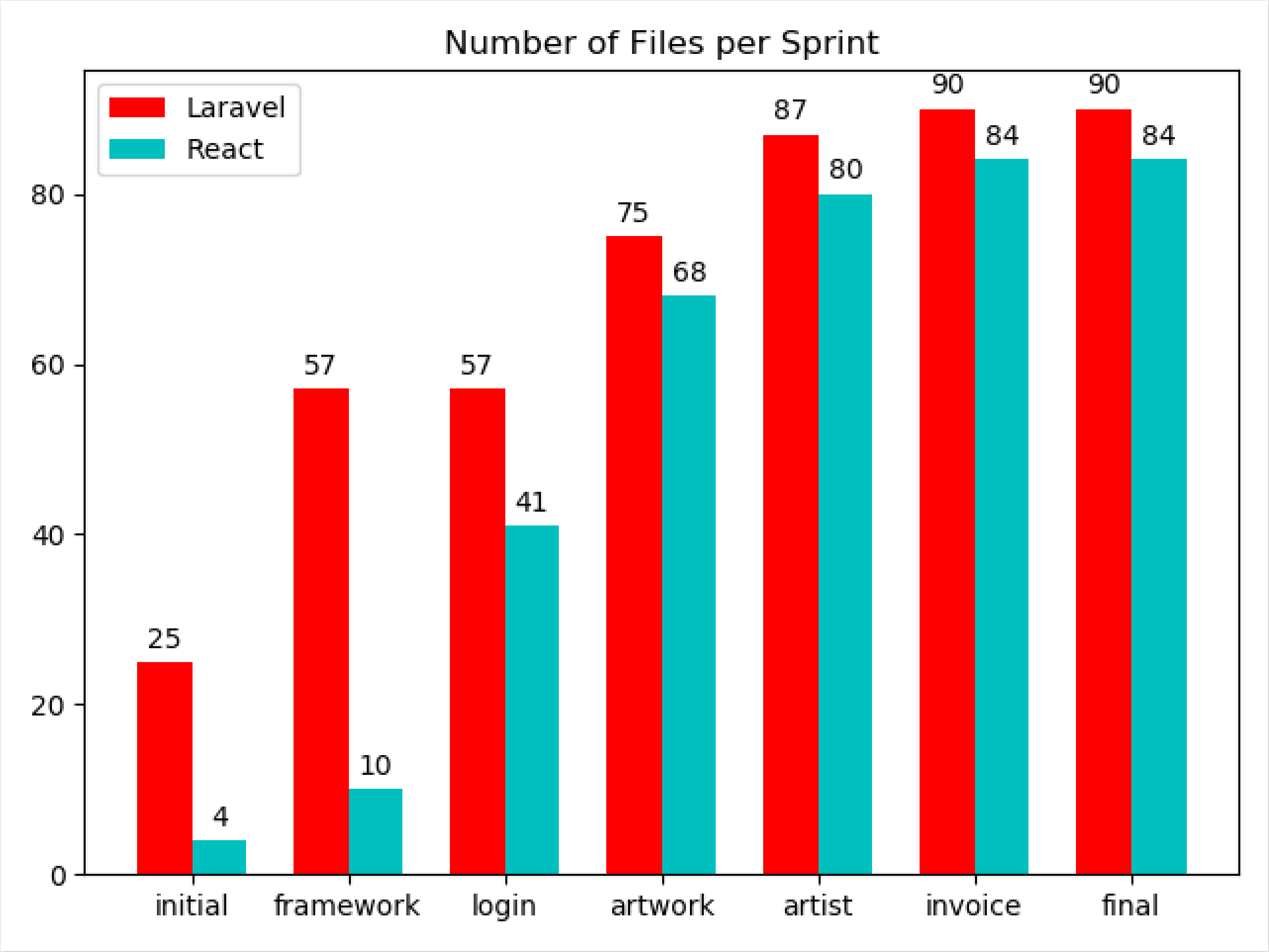
EFFERENT - COUPLING



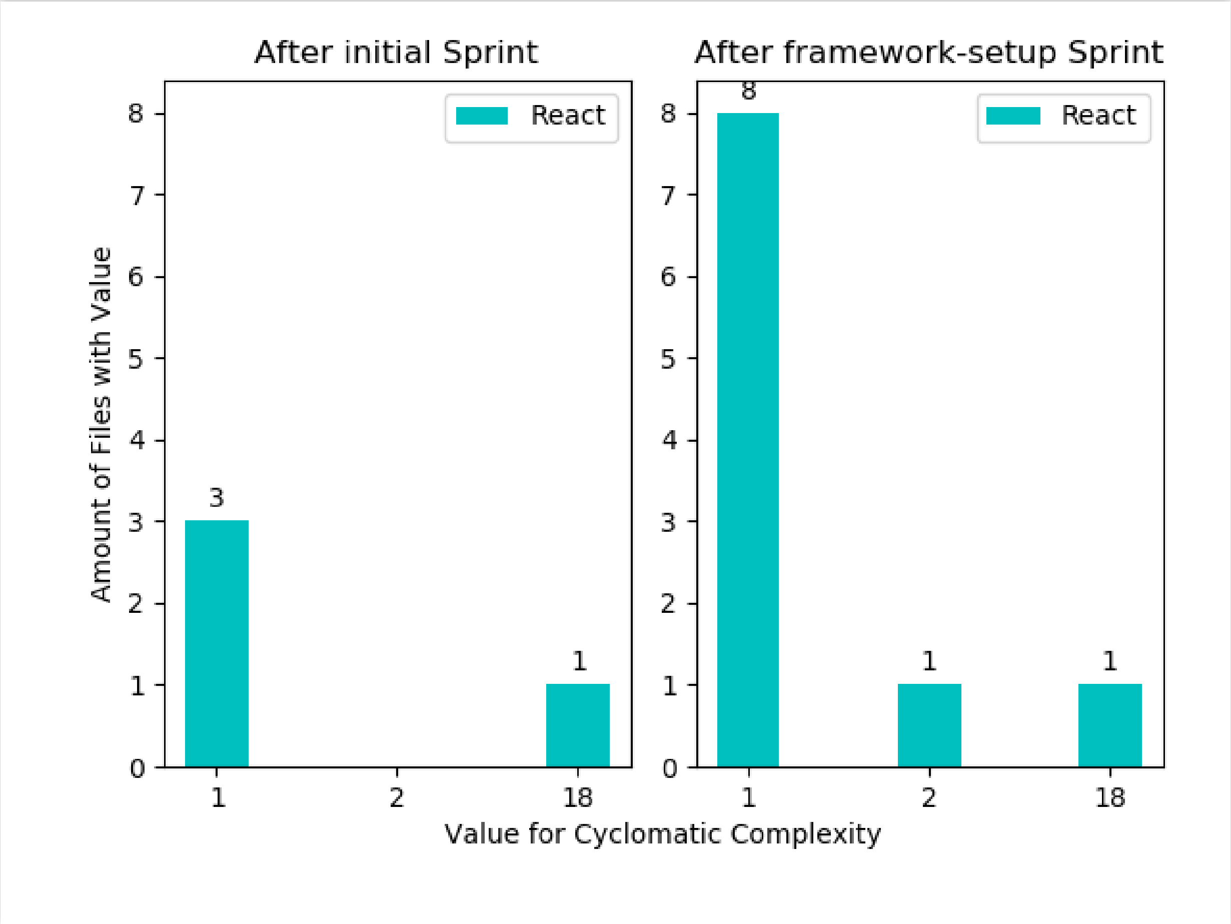
INSTABILITY



NUMBER OF FILES



CYCLOMATIC COMPLEXITY – JUMP



ATTRIBUTES AND EXPECTED IMPACT

ReactJS

- High degree of separation of concerns
  - × High Maintainability Index per file
  - × High Maintainability overall
- High interdependency
  - ✓ High values for the coupling measures
- Little back end functionality out of the box
  - ✓ High values for LLOC

Laravel

- Back end functionalities out of the box
- “Do more with less”
  - ✓ Low values for LLOC

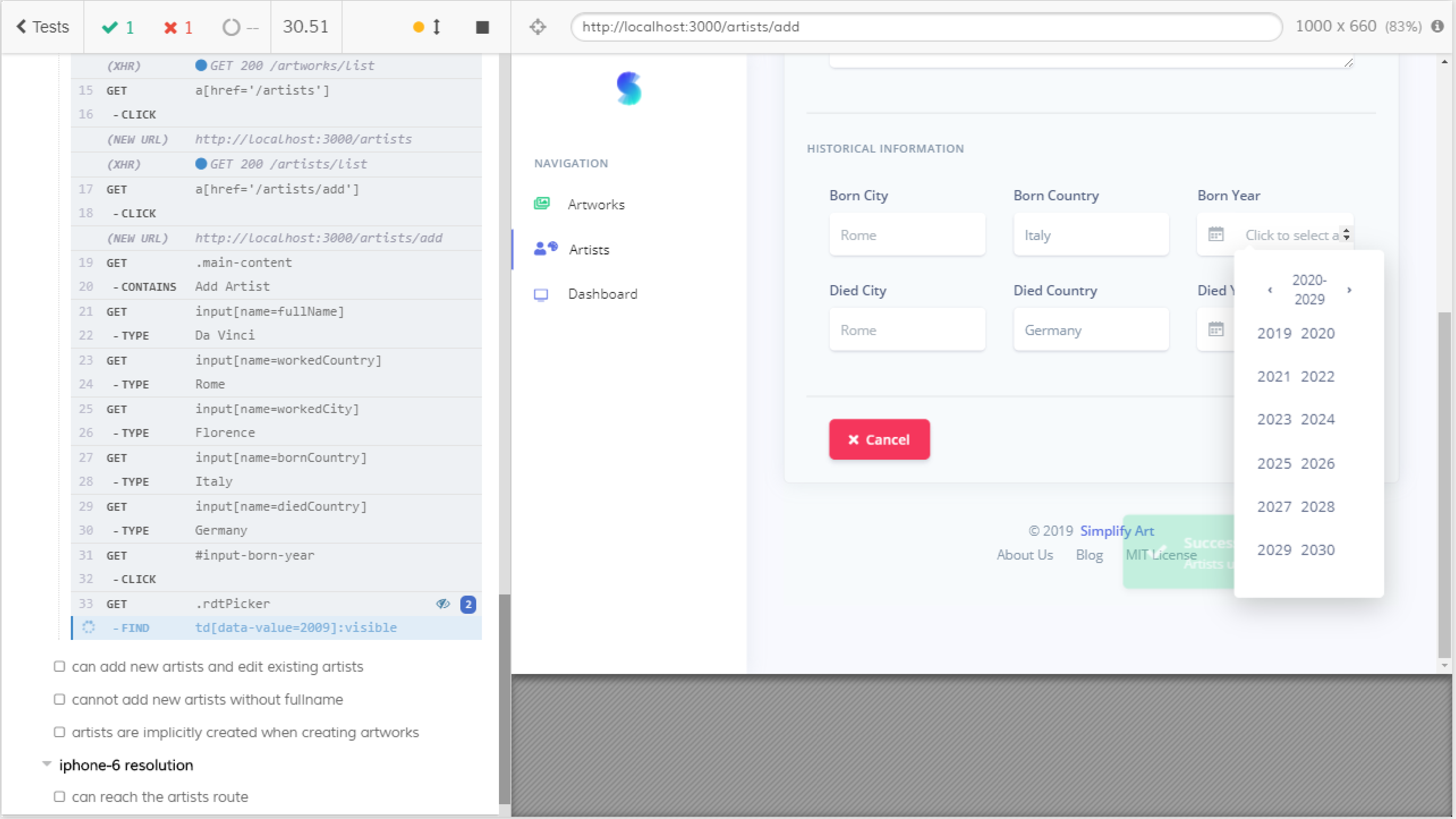
Development

SPRINTS

- initial
  - Bare-bones setup, “Hello World” page
- framework-setup
  - Add design template
- login
  - Authentication functionality
- artwork
  - Add, edit, delete artworks
- artists
  - Add, edit, delete artists
- invoice
  - Automatically generate PDF
- final-changes
  - Refactoring
  - Add dashboard

TESTING – CYPRESS

TESTING – CYPRESS – SCREENSHOTS



```
    (XHR)        ● GET 200 /artworks/list
15  GET          a[href='/artists']
16   - CLICK
    (NEW URL)    http://localhost:3000/artists
    (XHR)        ● GET 200 /artists/list
17  GET          a[href='/artists/add']
18   - CLICK
    (NEW URL)    http://localhost:3000/artists/add
19  GET          .main-content
20   - CONTAINS  Add Artist
21  GET          input[name=fullName]
22   - TYPE      Da Vinci
23  GET          input[name=workedCountry]
24   - TYPE      Rome
25  GET          input[name=workedCity]
26   - TYPE      Florence
27  GET          input[name=bornCountry]
28   - TYPE      Italy
29  GET          input[name=diedCountry]
30   - TYPE      Germany
31  GET          #input-born-year
32   - CLICK
33  GET          .rdtPicker                              2
     - FIND      td[data-value=2009]:visible
```

□ can add new artists and edit existing artists

□ cannot add new artists without fullname

□ artists are implicitly created when creating artworks

▾ iphone-6 resolution

□ can reach the artists route

HISTORICAL INFORMATION

Born City	Born Country	Born Year
Rome	Italy	Click to select a

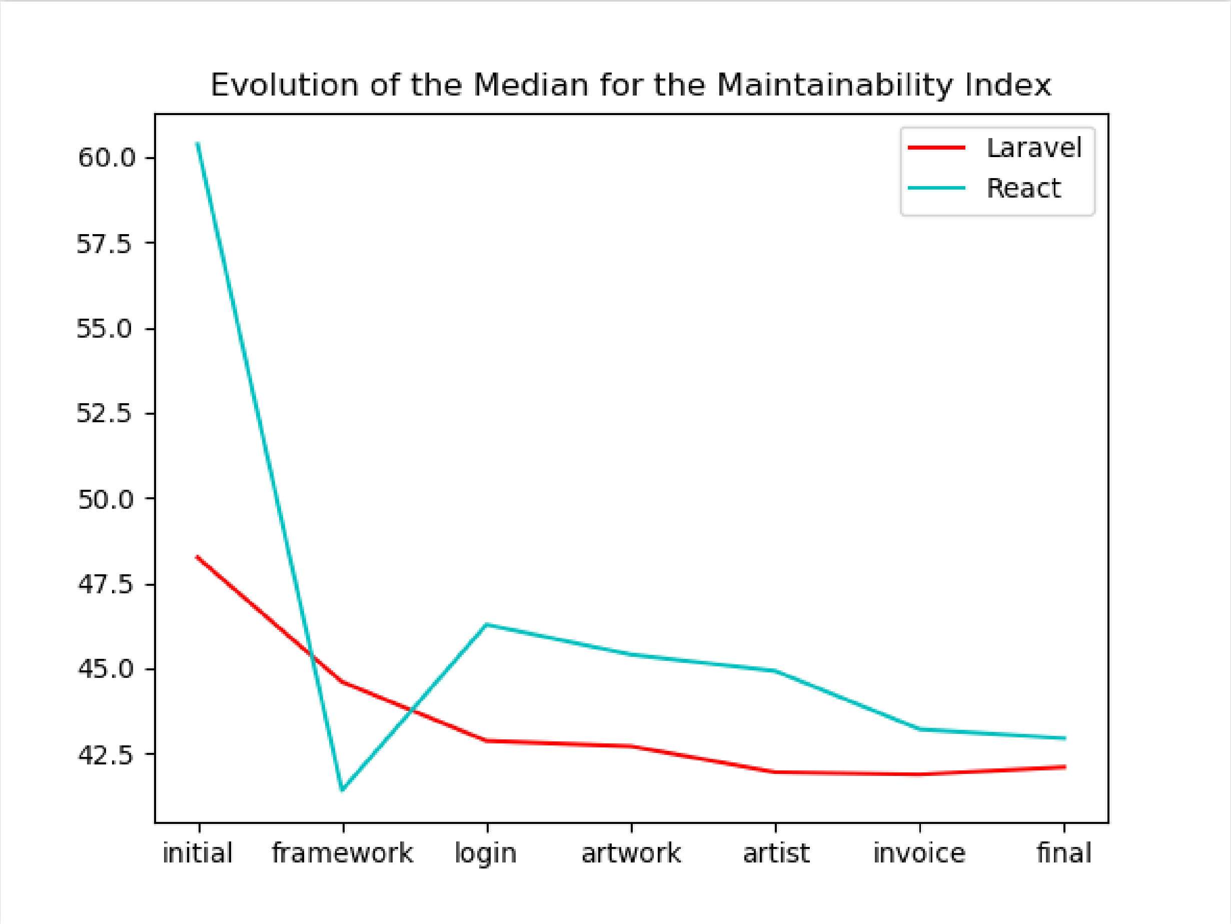
Died City	Died Country	Died Y
Rome	Germany	

Measurement Theory

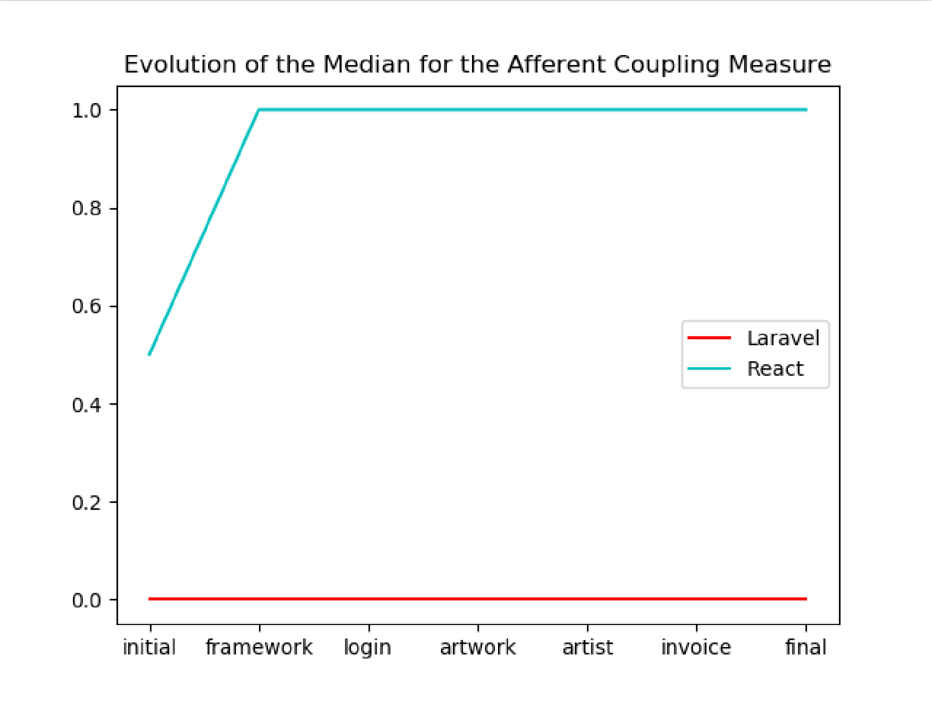
MEASUREMENTS AND SCALES

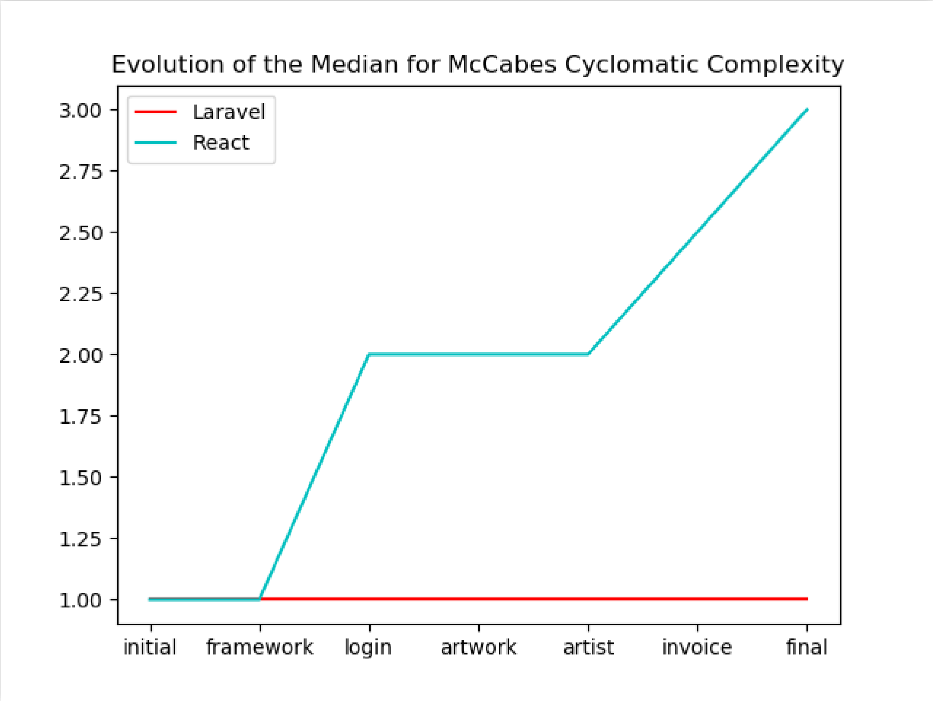
- Empirical relational system – $\mathcal{A}$
  - Entities and properties we observe
- Formal relational system – $\mathcal{B}$
  - Mathematical models
    - Numbers
    - Vectors
- Measurement – μ
  - Empirical object → formal object
- Scale – $(\mathcal{A}, \mathcal{B}, \mu)$
  - Mapping from an empirical relational system to a formal relational system
- Generally assume ordinal scale

MAINTAINABILITY INDEX – MEDIAN



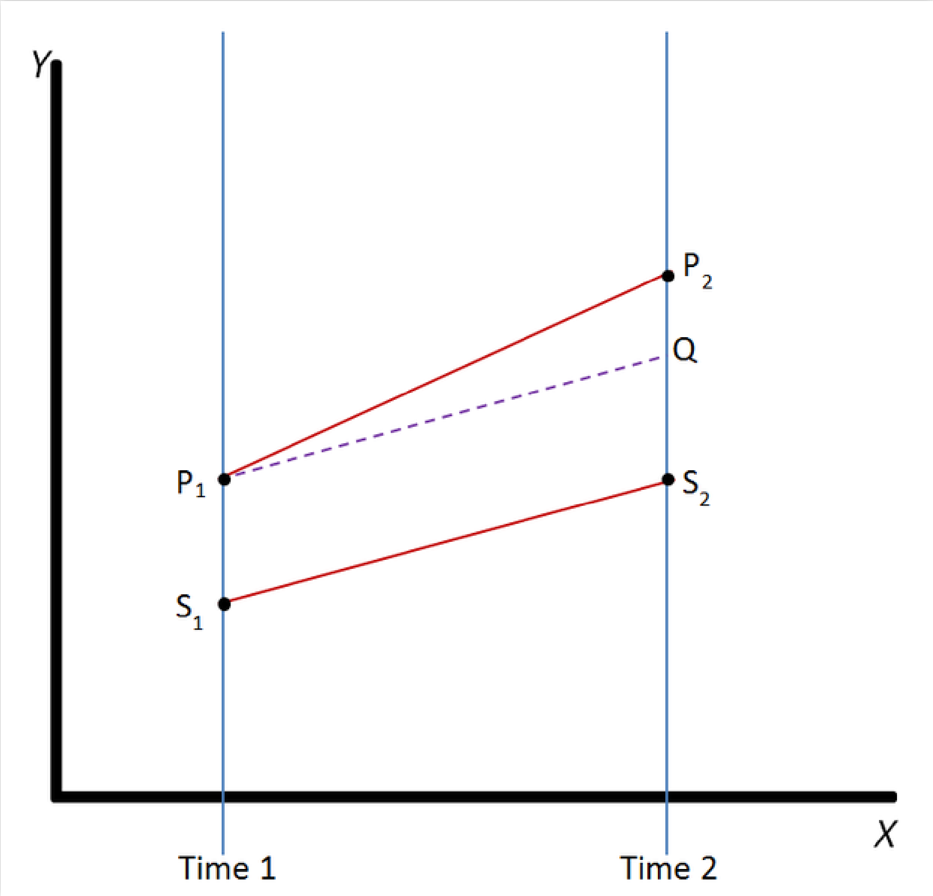
MEDIAN – GENERAL





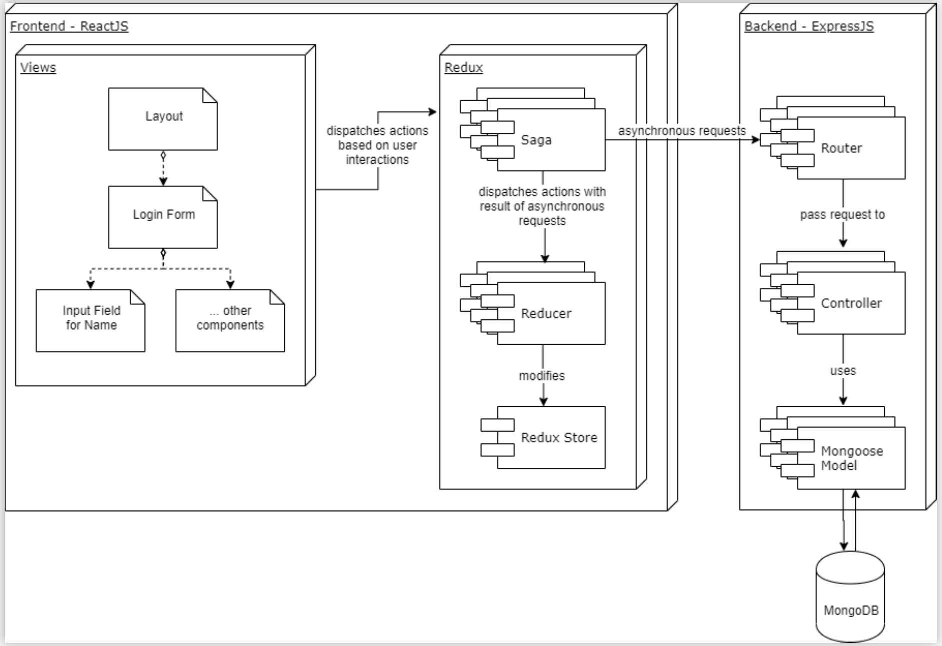
DIFFERENCE IN DIFFERENCES

- Would have addressed
  - Consistency issues
  - Differences in frameworks / programming languages
- Out of scope

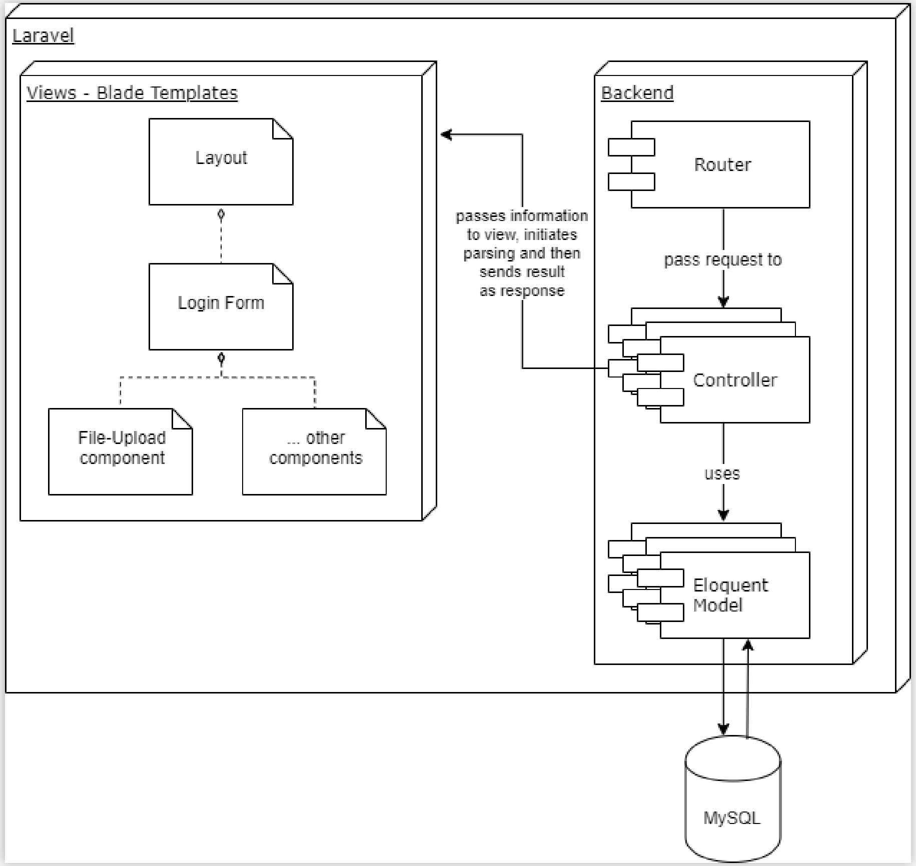


ARCHITECTURES

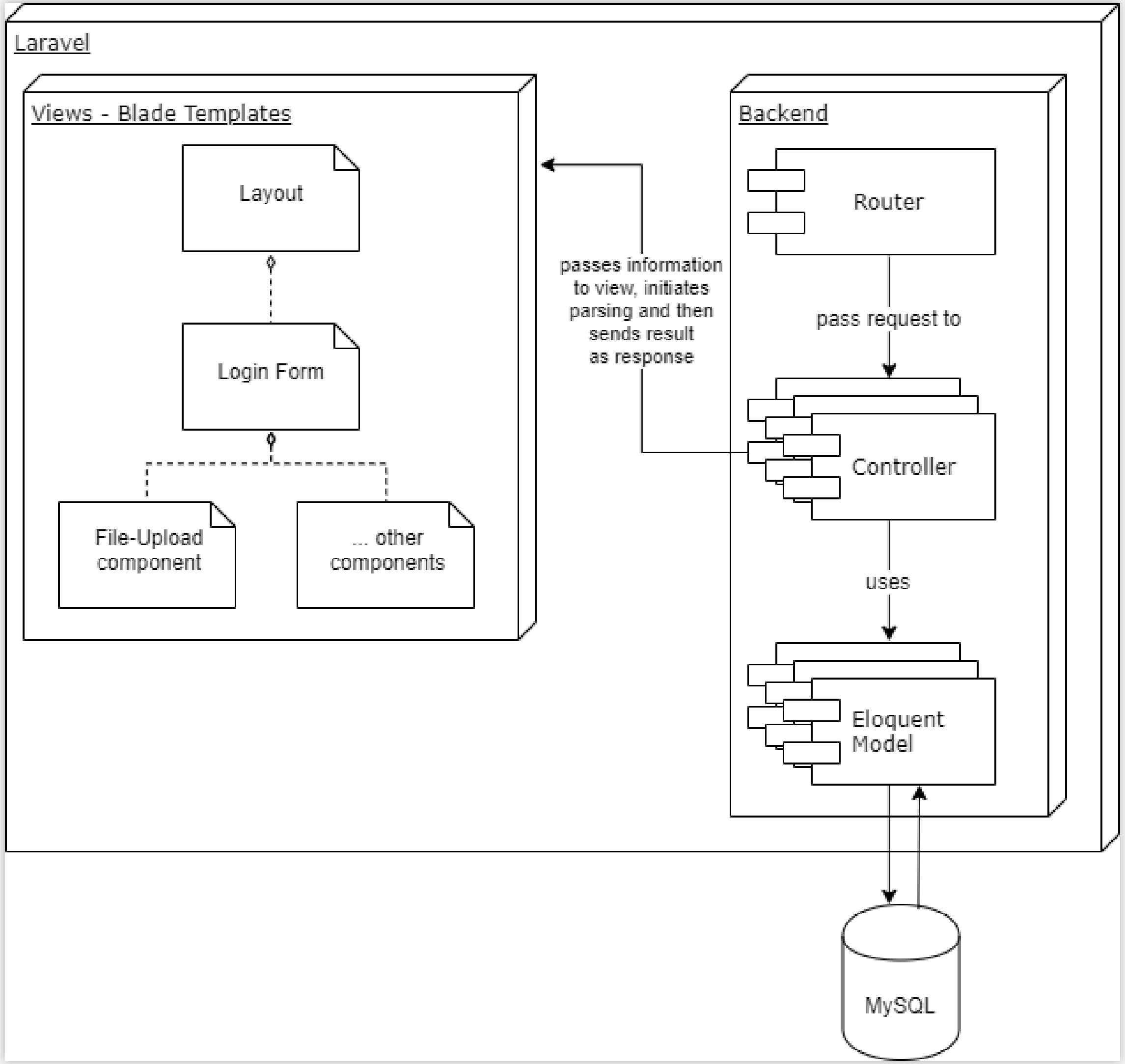
ReactJS



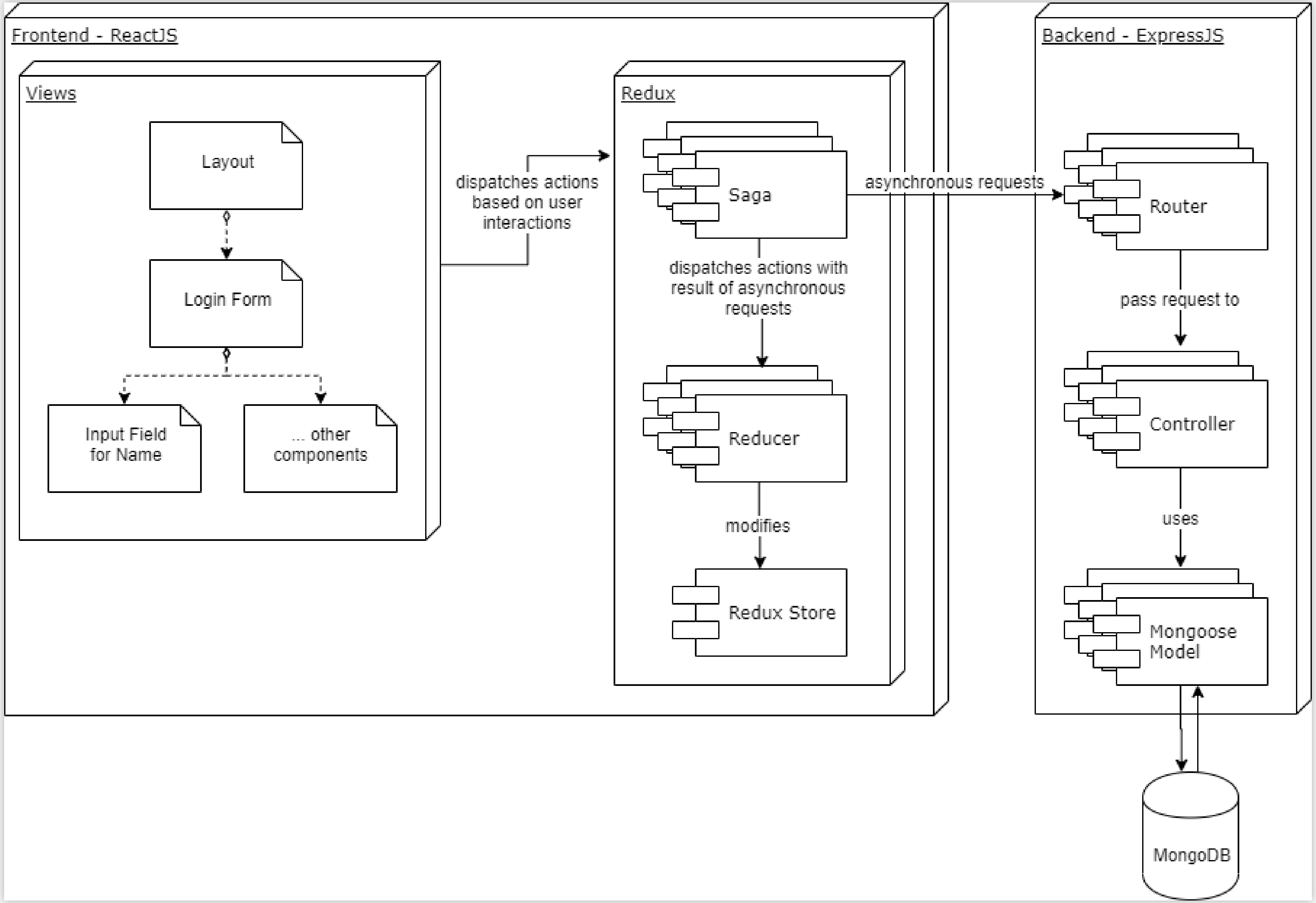
Laravel



LARAVEL – ARCHITECTURE



REACTJS – ARCHITECTURE



GENERAL RECOMMENDATION

ReactJS

- **Easy** to **setup**
- Developer **comfort**
  - react-scripts
- Sufficient debug properties
- **Separation of concerns** by design
- npm
- Deployment cheap and quick*

Laravel

- **Moderate effort** to setup correctly
- Developer comfort
  - Laravel Mix
  - Browsersync
- Relatively **cumbersome debugging**
- **Difficult deployment**
- **Substantial amount** of back end functionalities **out of the box**
- “Do more with less” by design

TAKE AWAYS

- Software Measurements can **aid the development process**
  - Highlight **core components** of architecture
  - Highlight **error prone** parts of the software
- If tools are available, using them is **easy**
- **Applicability** to framework comparison **questionable**
- There is room for work
  - **Precise definitions** of software measurements
  - **Tools** to calculate them