



using R and shiny at FRISS

satRday conference, Budapest, sep 3 2016

F R I S S
fraud, risk & compliance





FRISS score®



Herman Sontrop
Ph.D

Junior scientist



Erwin Schuijftvlot
M.Sc

Senior data scientist



Sebastiaan de Vries
M.Sc

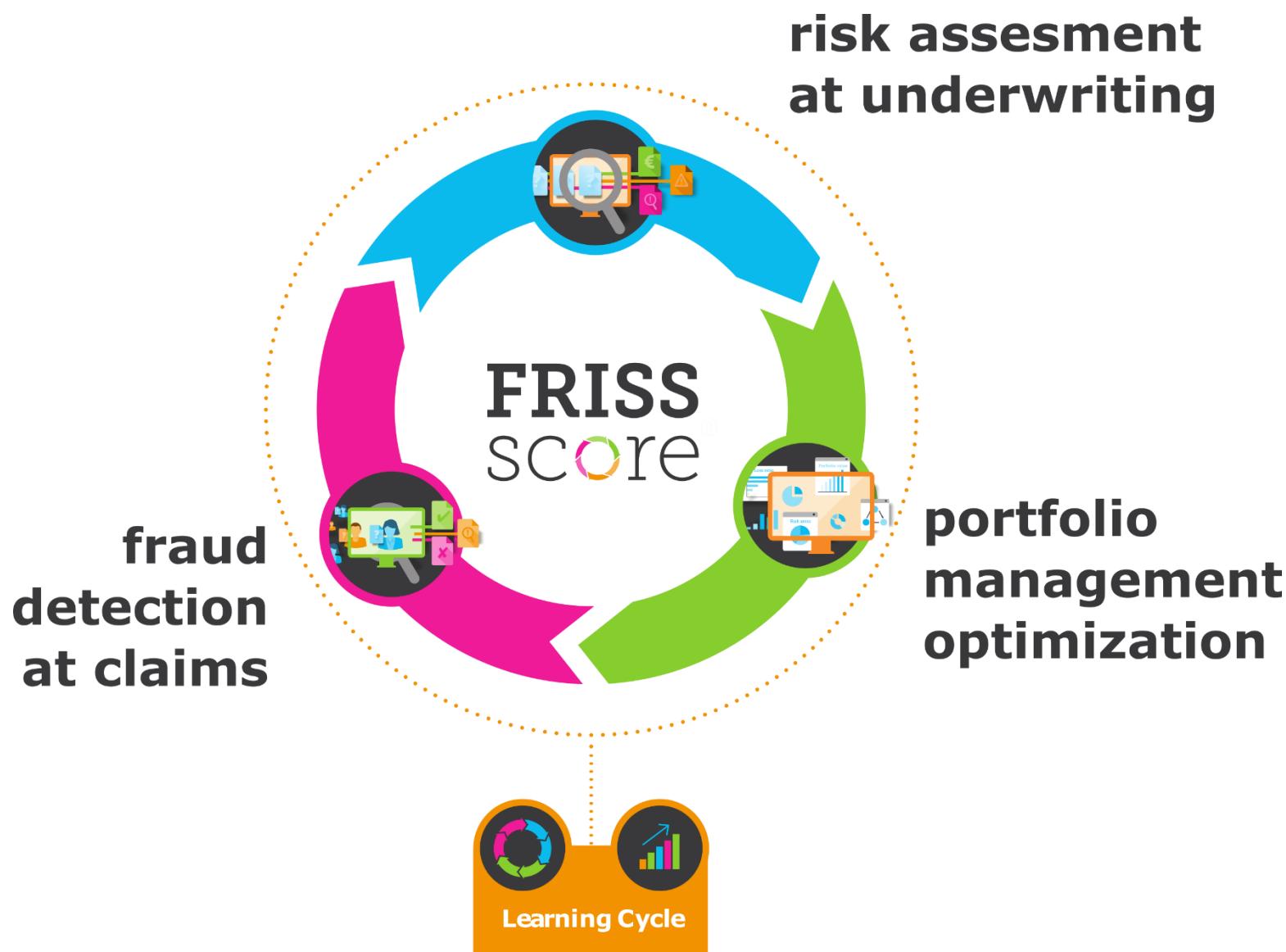
Senior data scientist



Gian Luigi Chiesa
M.Sc

Junior data scientist





How does R help us to
detect **risk**, eliminate **fraud** &
secure **compliance**?



F R I S S
fraud, risk & compliance





Analytics



Text Mining

Predictive Underwriting



Device screening



Repair shop Screening



Image Screening

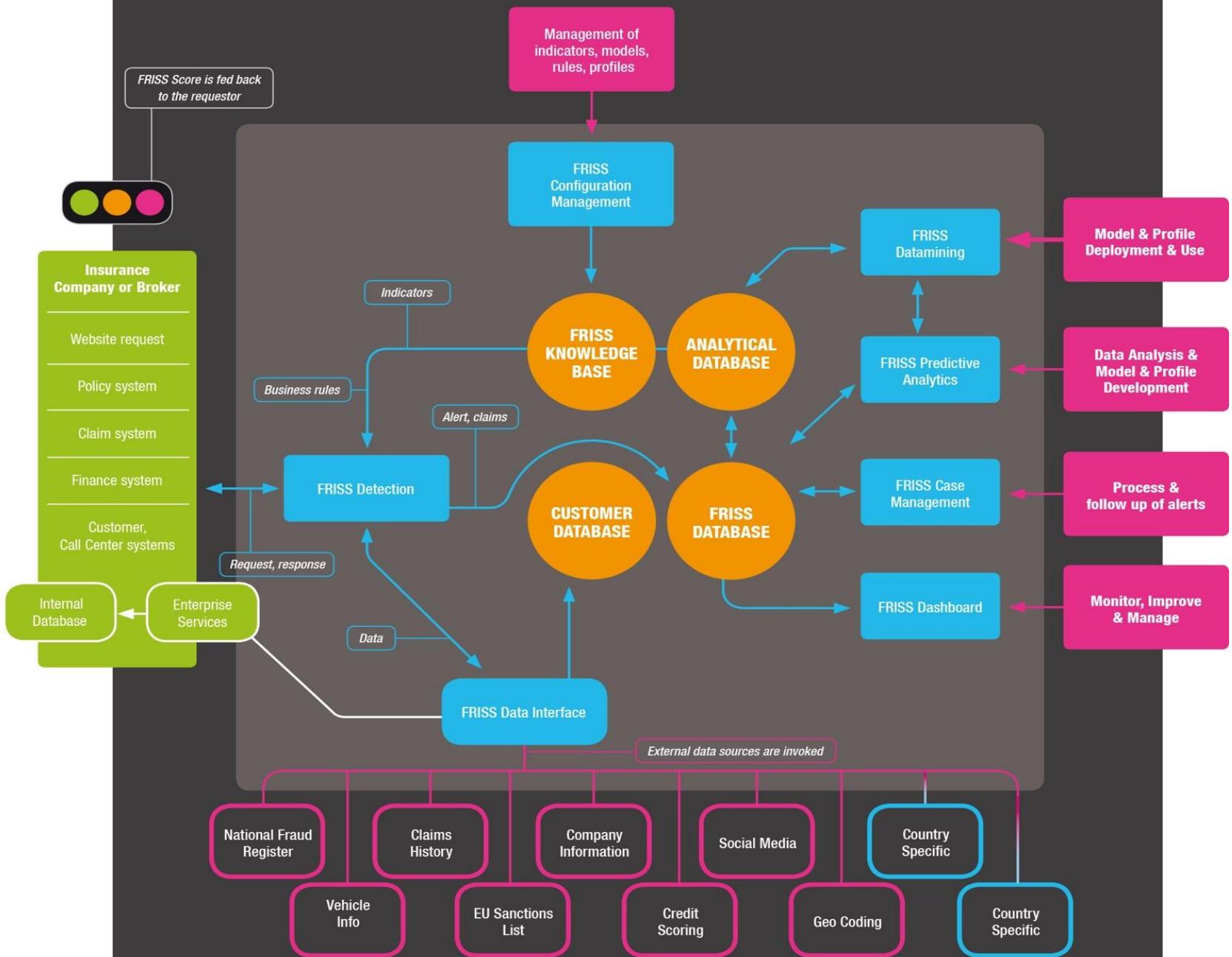


Agent profiling



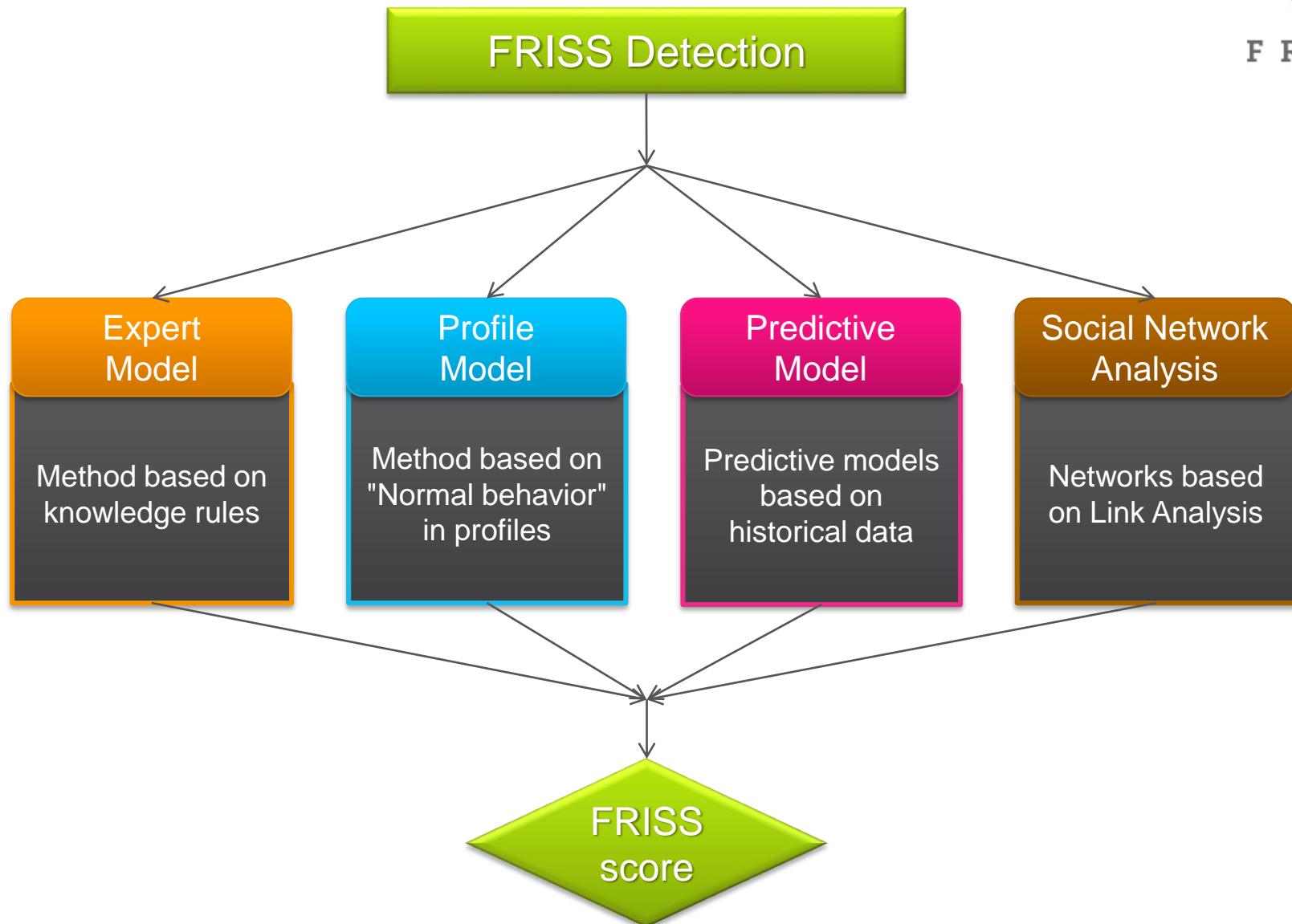
Social Network Analysis

FRISS Platform





F R I S S





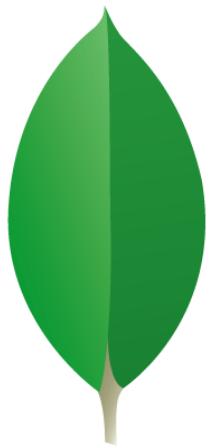
F R I S S
fraud, risk & compliance

What tools do we use?

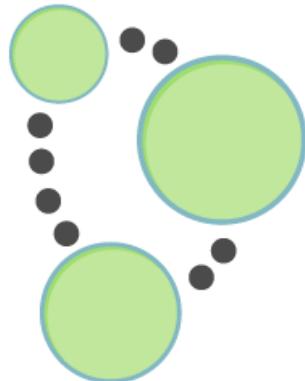


Main app building blocks

- R
- shiny
- shiny server, docker, apache
- javascript
- external databases & tools (next slide)
- **modules**
<http://shiny.rstudio.com/articles/modules.html>
- **htmltemplates**
<http://shiny.rstudio.com/articles/templates.html>
- **htmlwidgets**
<http://www.htmlwidgets.org/>



mongoDB



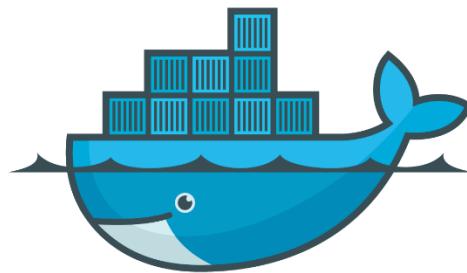
Neo4j



elastic



Microsoft®
SQL Server®



docker

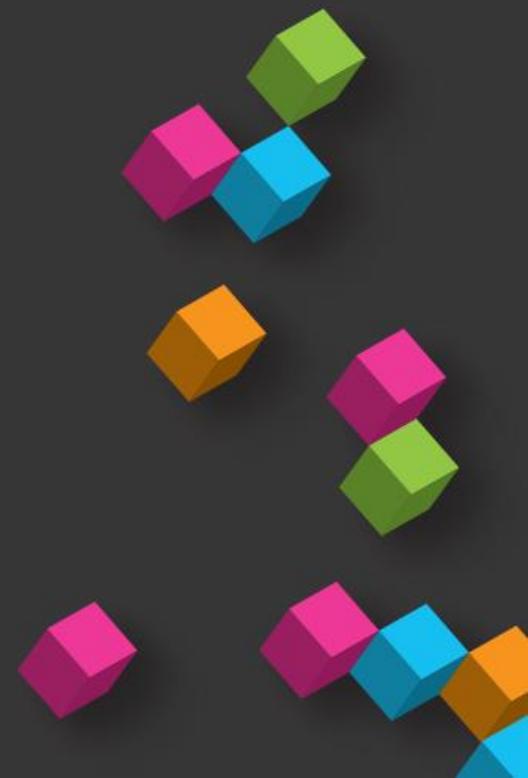


F R I S S
fraud, risk & compliance

To paraphrase a quote of a quote:

**Everything that can be implemented in JavaScript,
one day will be implemented in JavaScript**

Gábor Csárdi <http://budapest.satrdays.org/>



HTML

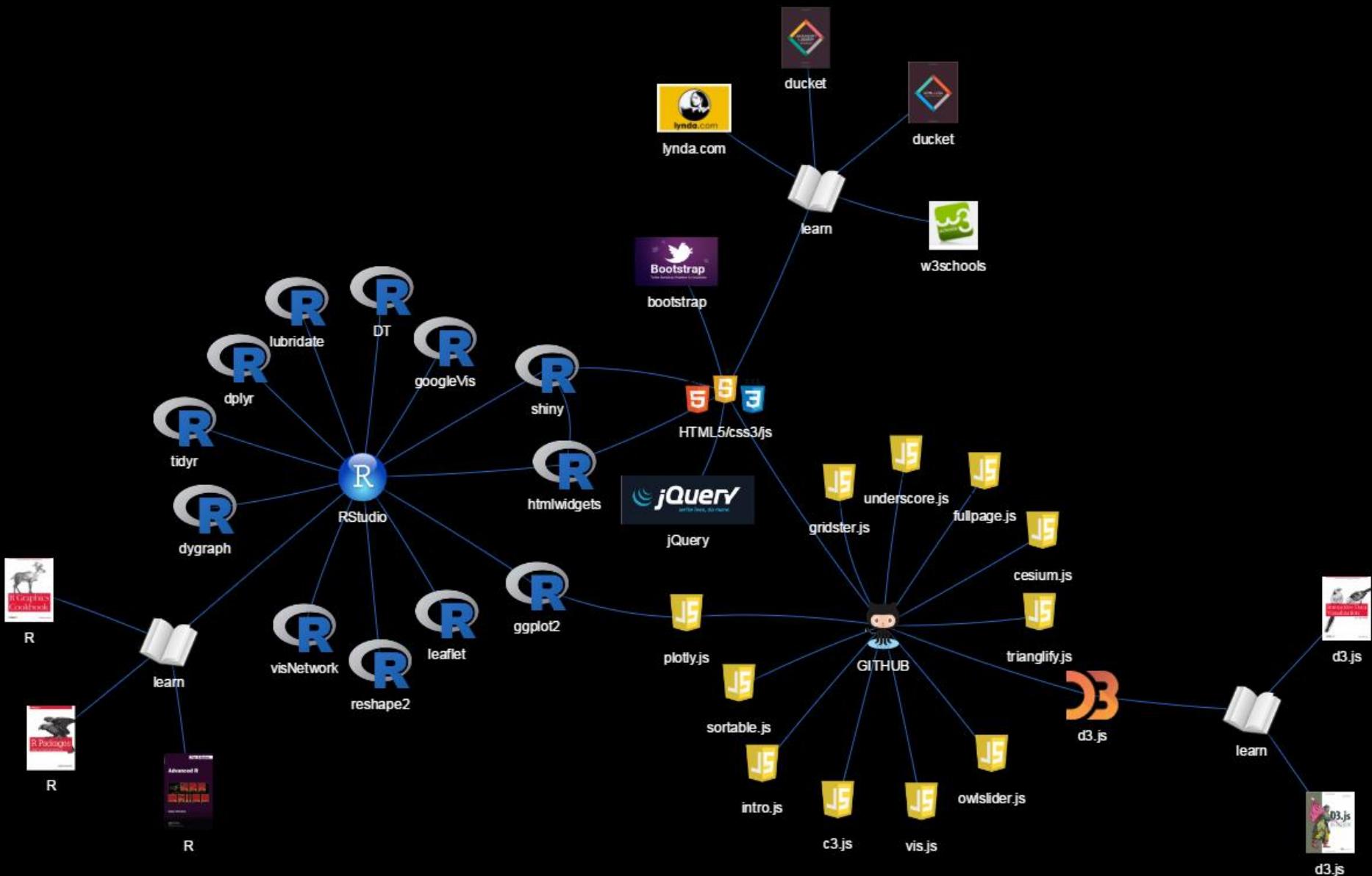


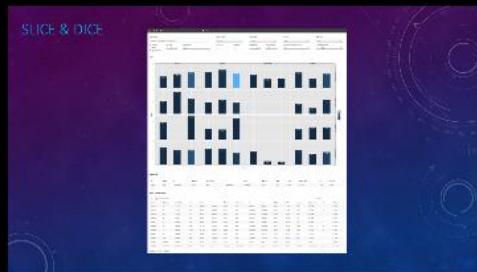
JS



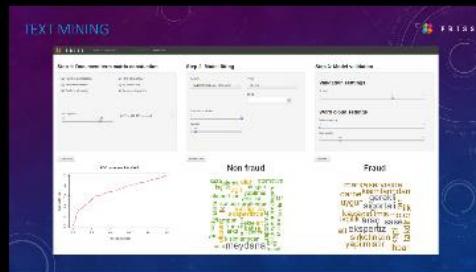
CSS



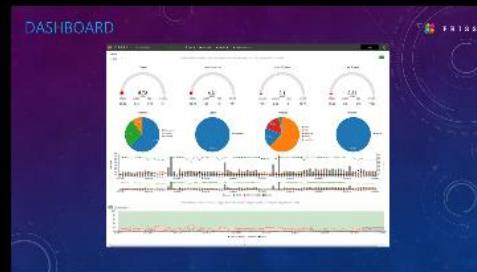




Slice & Dice
Friss Analytics



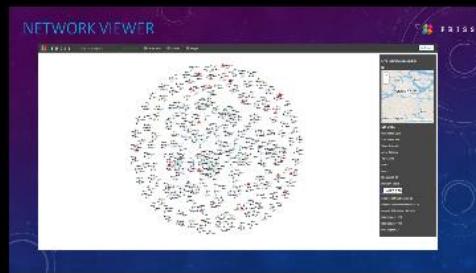
Text Mining
Friss Analytics



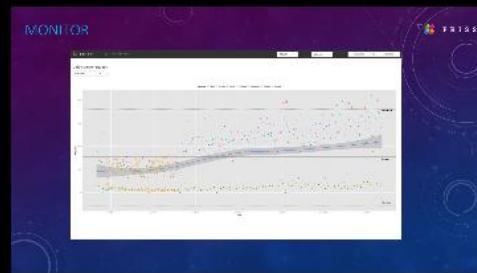
Dashboard
Friss Analytics



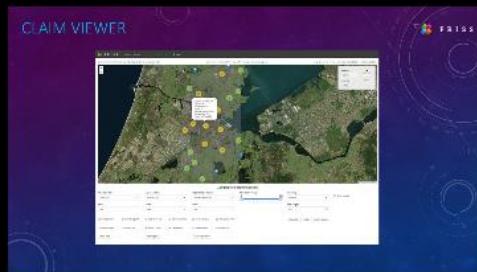
Model Fitting
Friss Analytics



Network Viewer
Friss Analytics



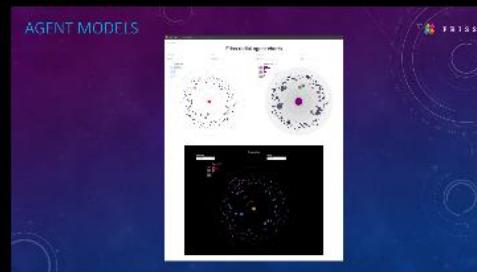
Monitor
Friss Analytics



Claim Viewer
Friss Analytics



Claim Dashboard
Friss Analytics

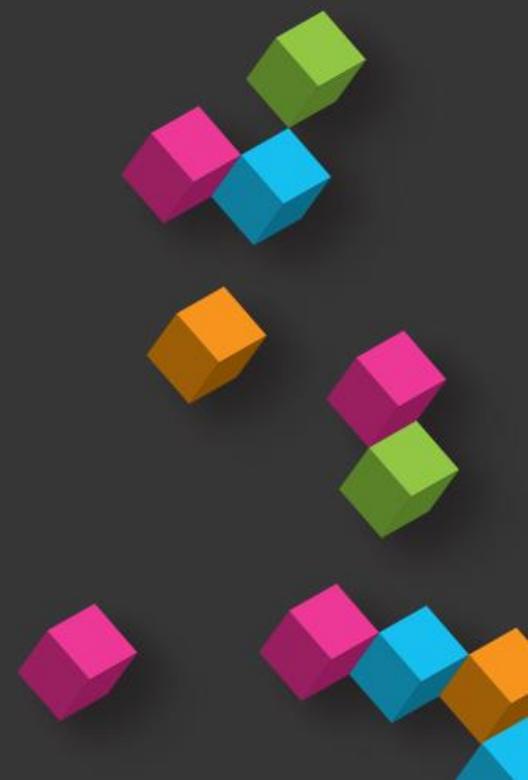


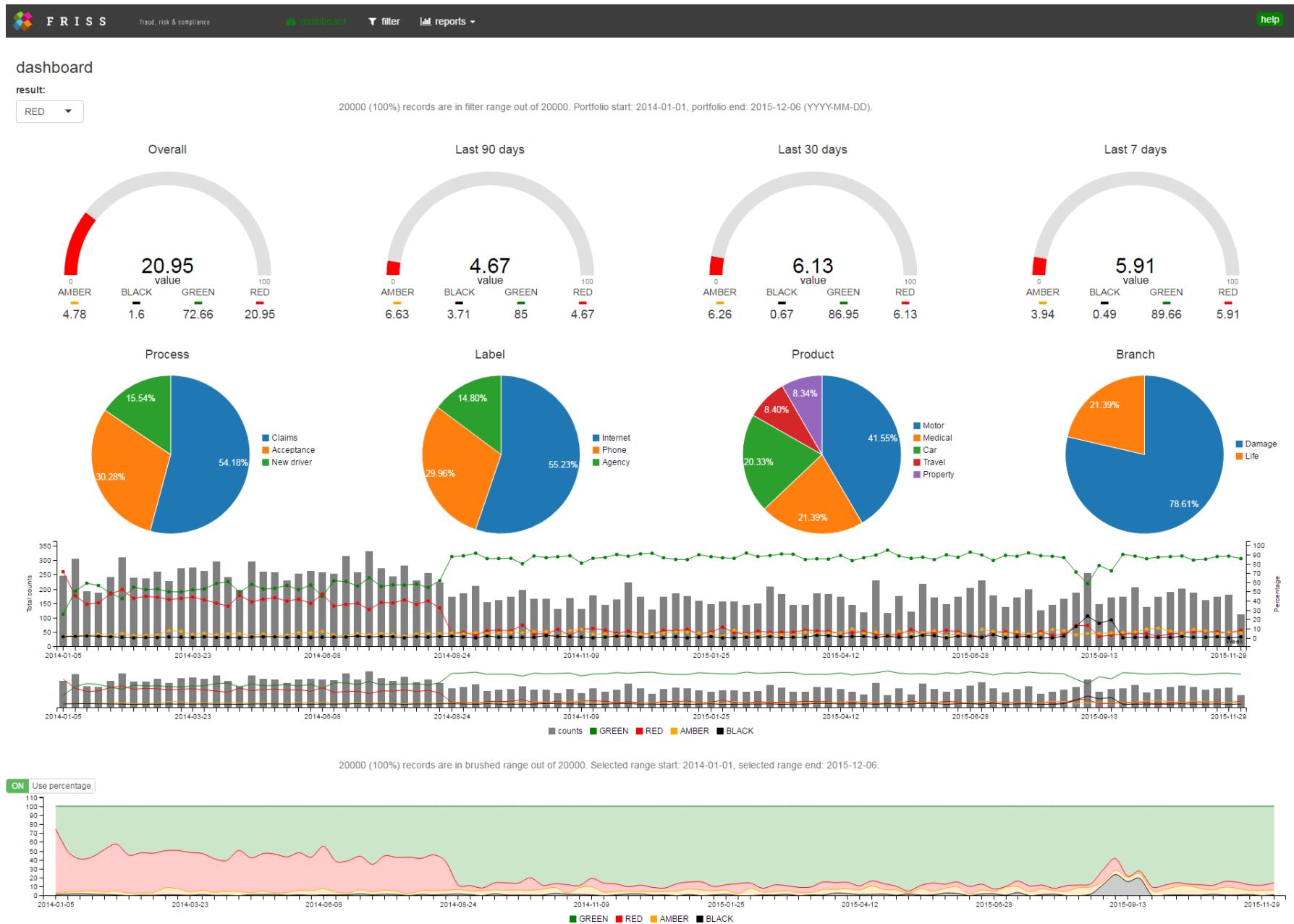
Agent Models
Friss Analytics



shiny modules

F R I S S
fraud, risk & compliance



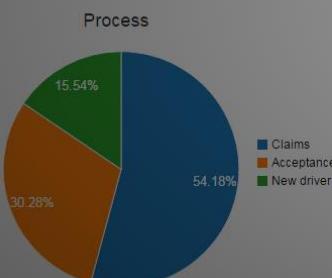
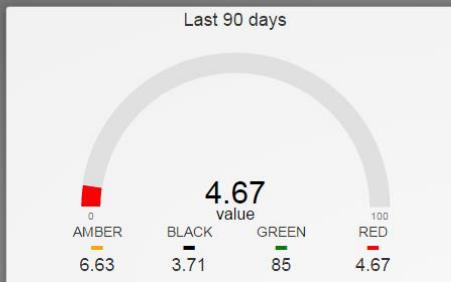


dashboard

result:

RED

20000 (100%) records are in filter range out of 20000. Portfolio start: 2014-01-01, portfolio end: 2015-12-06 (YYYY-MM-DD).

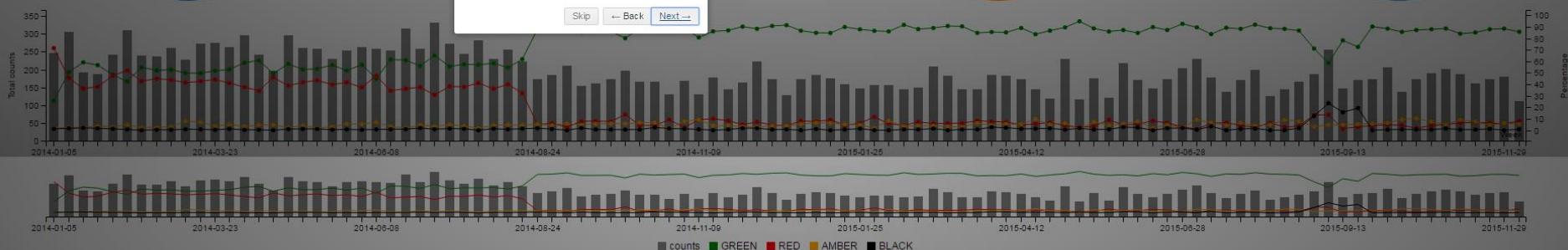
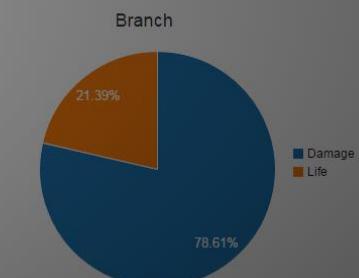
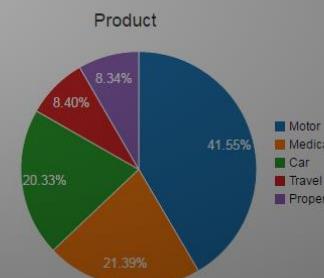


The gauges show the percentage of alerts of a given alert status over a fixed time period.

For instance, this gauge tells you that over the last 3 months, **4.67%** of the alerts were of result type **RED**.

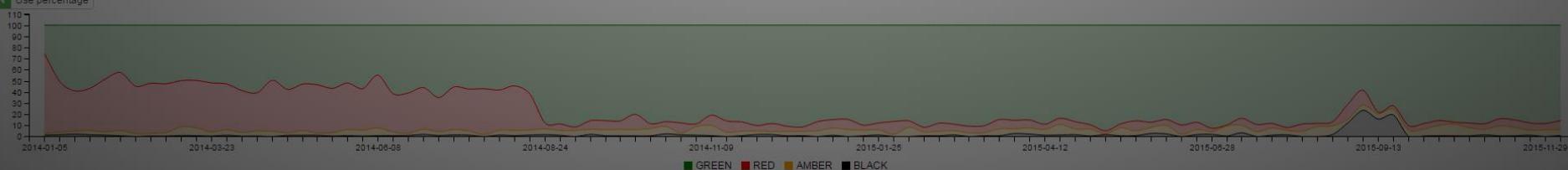
The two gauges on the right provide the same information, however, taken over the last 30 days and the last 7 days, while the most left gauge tells you the percentage taken over all alerts in the system.

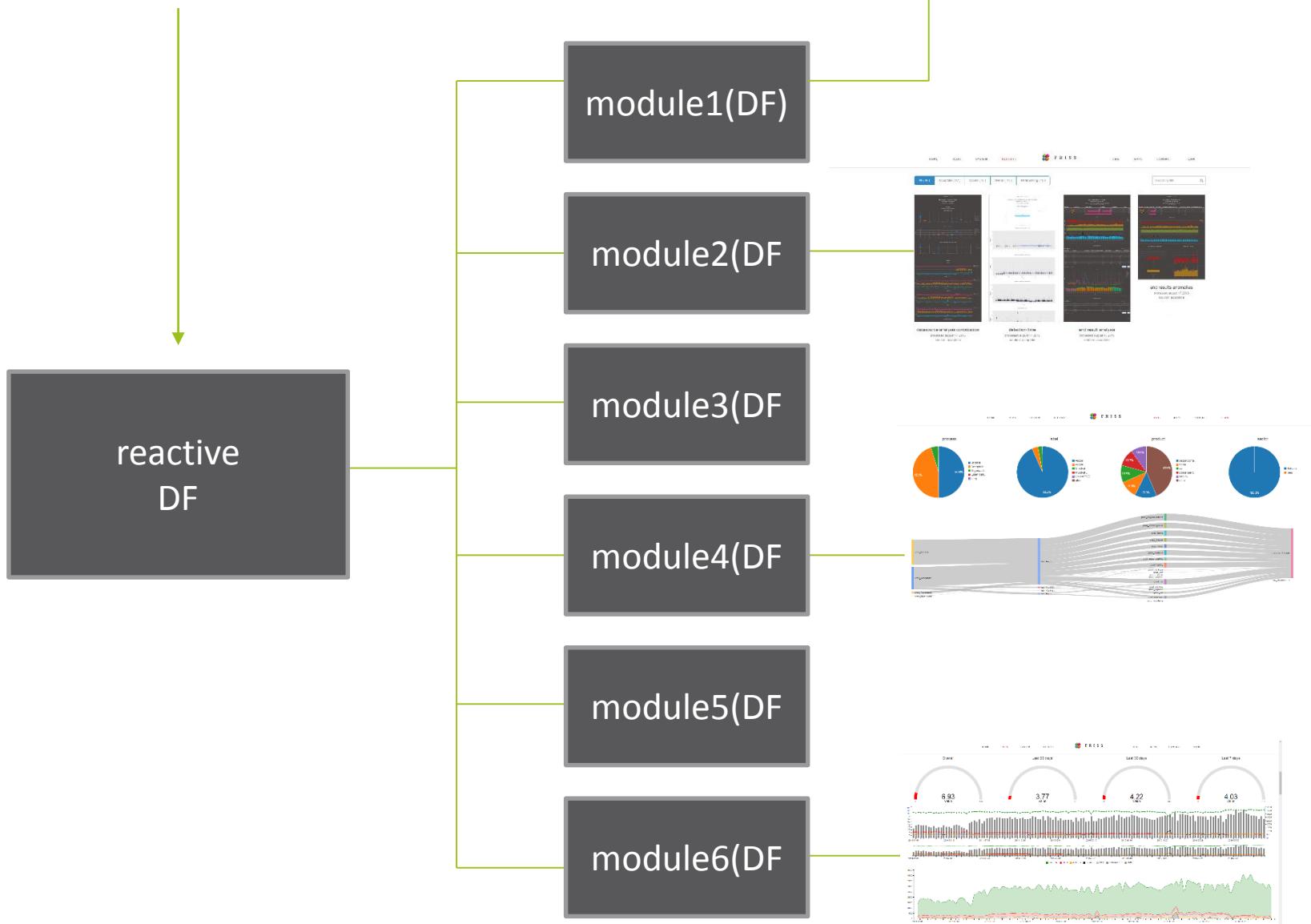
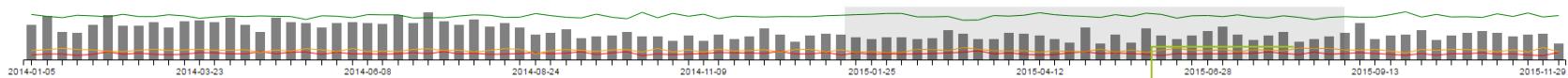
- Internet
- Phone
- Agency



20000 (100%) records are in brushed range out of 20000. Selected range start: 2014-01-01, selected range end: 2015-12-06.

ON Use percentage

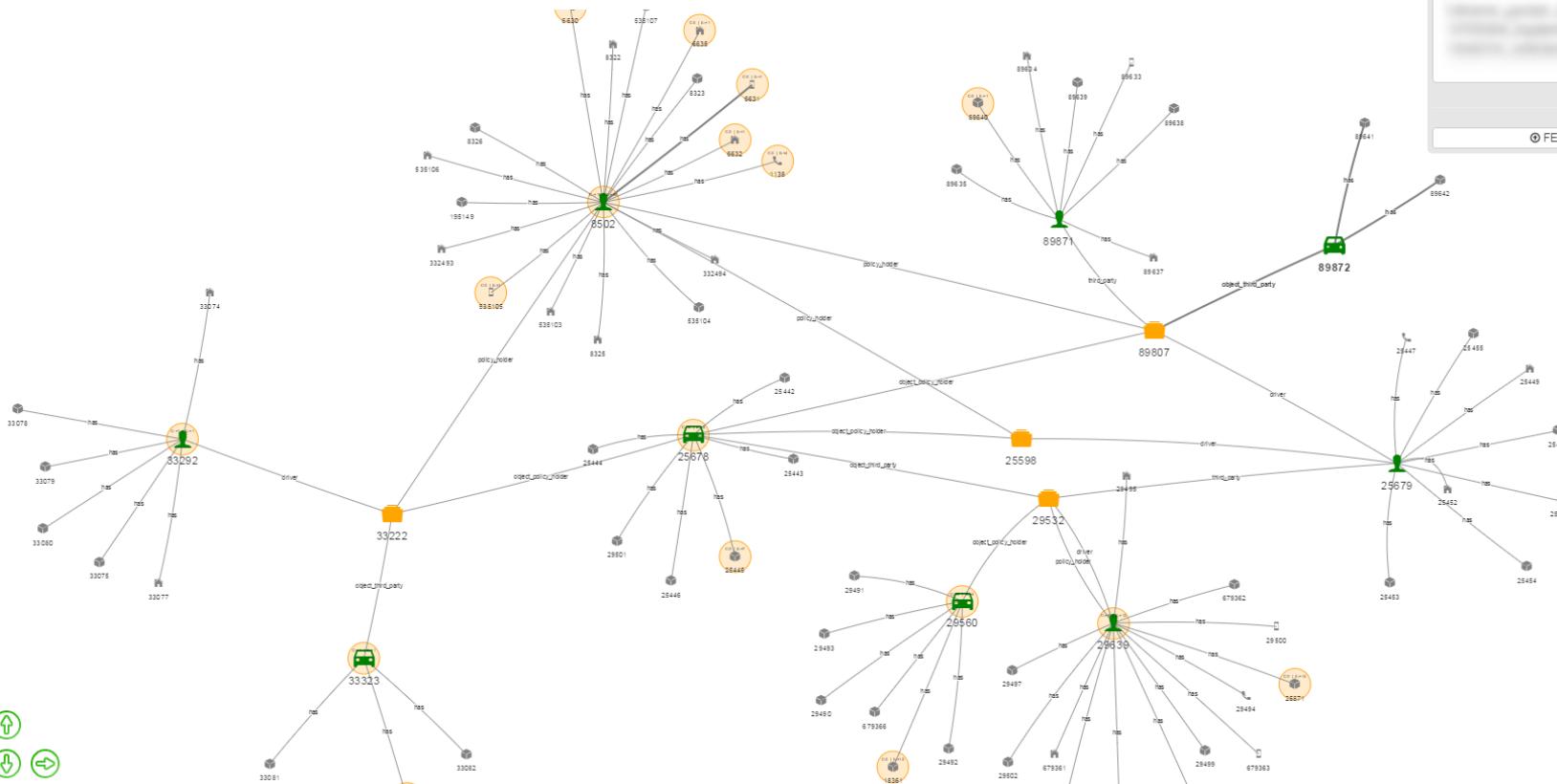




controls

manual ring2 ring3

FETCH NETWORK



timeline

2011-03-02

save network

2015-03-12



date type

occurred

label

id

type filter

Entity Claim Attribute

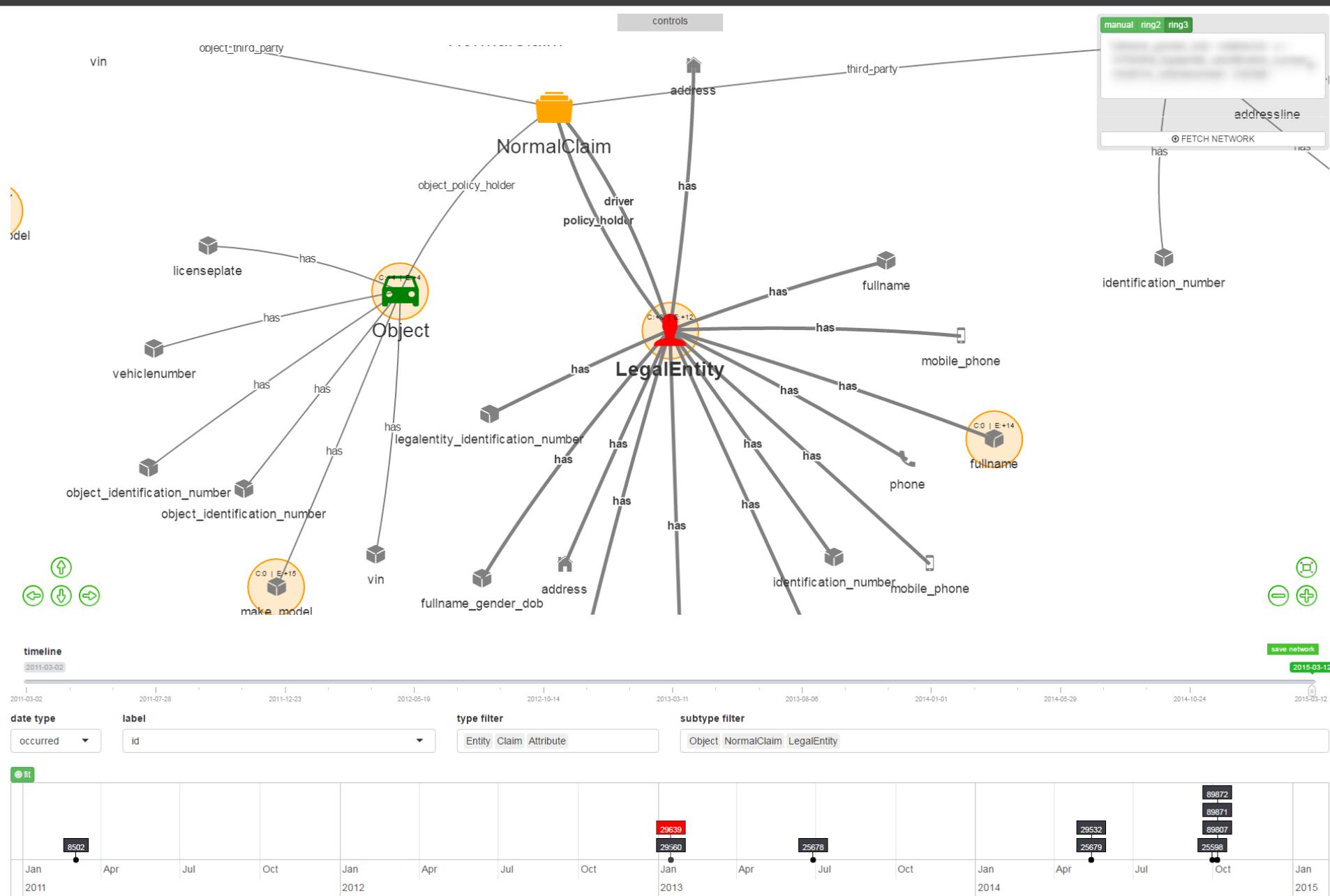
subtype filter

Object NormalClaim LegalEntity

2015-03-12

fit

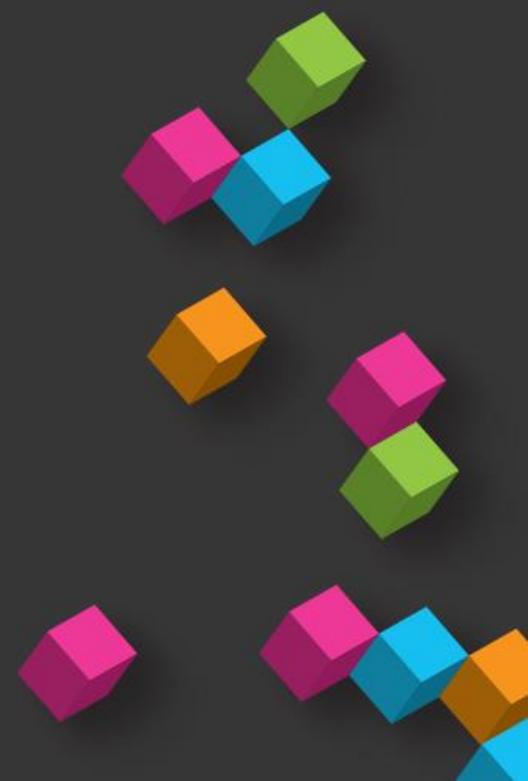






F R I S S
fraud, risk & compliance

Me too! Where do I start?

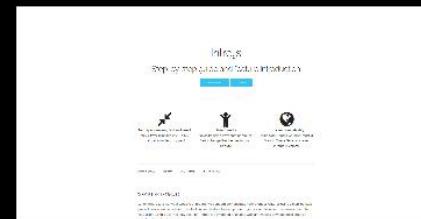




fullpage.js
beautiful fullscreen scrolling websites



gridster.js
drag-and-drop multi-column jQuery grids



intro.js
*step-by-step guide and feature introduction
for webpages*



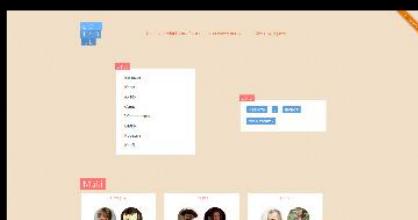
c3.js
D3-based reusable chart library



cesium.js
world-class WebGL 3D globes and maps



vis.js
*d3 based Timeline & Network charts for
large dynamic datasets*



sortable.js
*reorderable drag-and-drop lists for modern
browsers*



trianglify.js
algorithmically generated triangle art



owlslider.js
jQuery responsive carousel sliders

find cool stuff!!

d3 Data-Driven Documents



D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

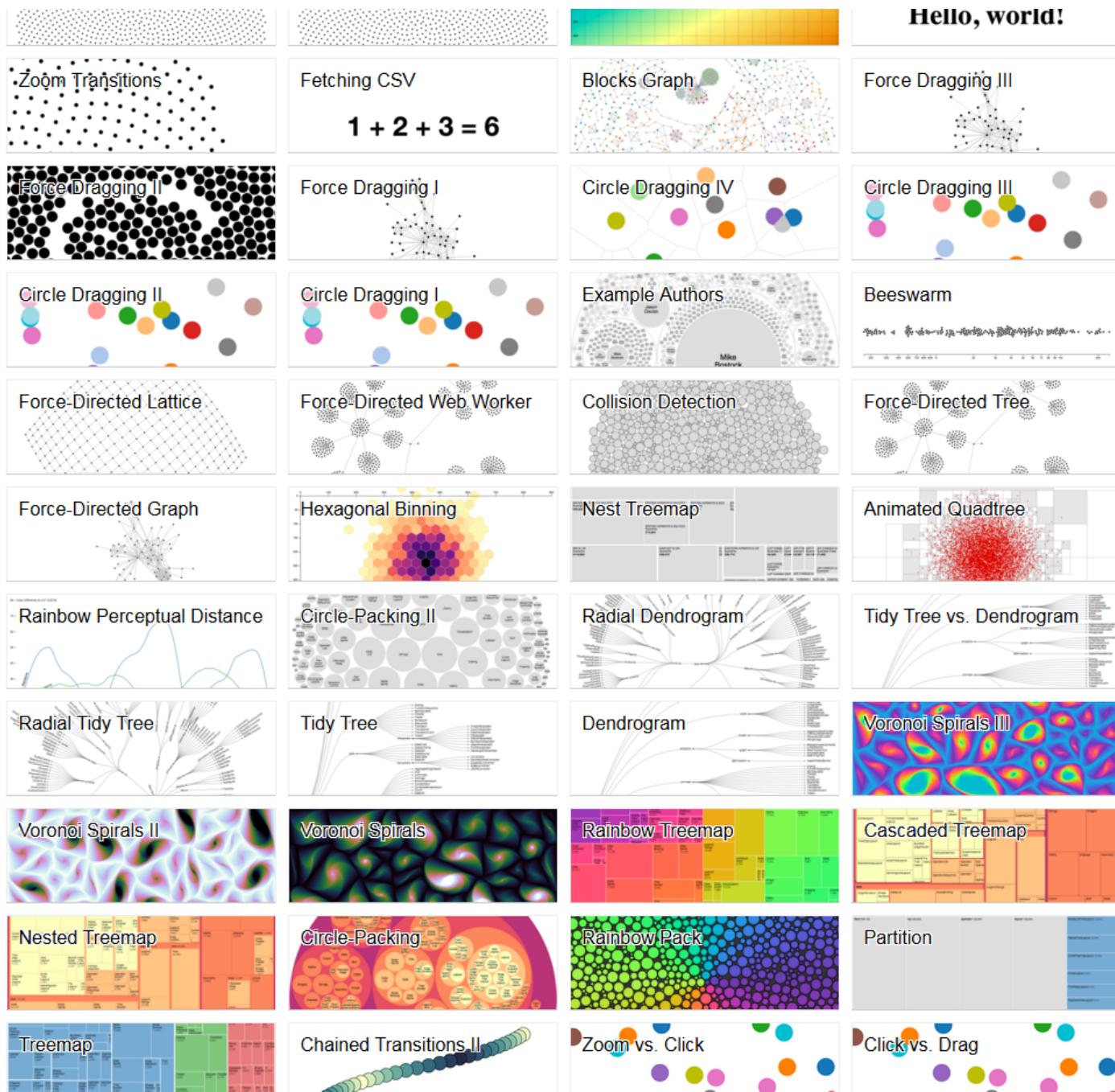
See [more examples](#).

Download the latest version (4.2.2) here:

- [d3.zip](#)

To link directly to the latest release, copy this snippet:

create HTMLWidgets e.g. using d3.js





F R I S S



Hadley Wickham
hadley

@rstudio
Houston, TX
h.wickham@gmail.com
<http://hadley.nz>
Joined on 1 Apr 2008

7.1k Followers **110** Starred **7** Following

Organizations

[Overview](#) [Repositories](#) [Public activity](#) [Follow](#)

Popular repositories

	ggplot	An implementation of the Grammar of Graphics in R	1,984 ★
	dplyr	Dplyr: A grammar of data manipulation	1,391 ★
	devtools	Tools to make an R developer's life easier	1,318 ★
	adv-r	Advanced R programming: a book	815 ★
	rvest	Simple web scraping for R	641 ★

5,196 contributions in the last year



Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Less More

ggplot2 2K stars



Mike Bostock
mbostock

I design tools for humans. Creator of [d3](#). Former @nytgraphics editor.

San Francisco, CA
mike@bostock.org
<http://bostock.org>
Joined on 25 Mar 2010

13.1k Followers **37** Starred **13** Following

Organizations



[Overview](#) [Repositories](#) [Public activity](#) [Follow](#)

Pinned repositories

	d3/d3	Bring data to life with SVG, Canvas and HTML.	54,374 ★
	d3/d3-geo	Geographic projections, spherical shapes and spherical trigonometry.	72 ★
	d3/d3-scale	Encodings that map abstract data to visual representation.	703 ★
	d3/d3-shape	Graphical primitives for visualization, such as lines and areas.	1,264 ★
	topojson	An extension to GeoJSON that encodes topology.	2,458 ★

5,908 contributions in the last year



Contribution activity

Period: **1 week** ▾

d3 54K stars

OVERVIEW

TUTORIAL

ARTICLES

GALLERY

REFERENCE

DEPLOY

HELP



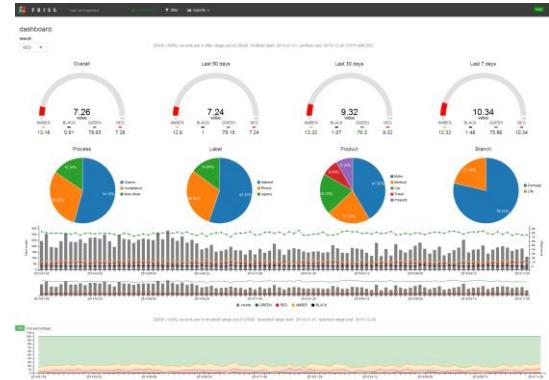
How to build a JavaScript based widget

Herman Sontrop

Erwin Schuijtvlot

2016-07-06

- Introduction
 - Creating a dashboard app
 - Dashboard data
 - Creating a dynamic help system
 - Creating your own input binding
 - Tutorial materials
- Stand alone examples
- Recipe to construct charts
- A first example
 - step 1
 - step 2
 - step 3
 - step 4
- Extending the gauge
- Creating the widget



OVERVIEW

TUTORIAL

ARTICLES

GALLERY

REFERENCE

DEPLOY

HELP

How to add functionality to JavaScript widgets

Herman Sontrop

Erwin Schuijtvlot

2016-07-06

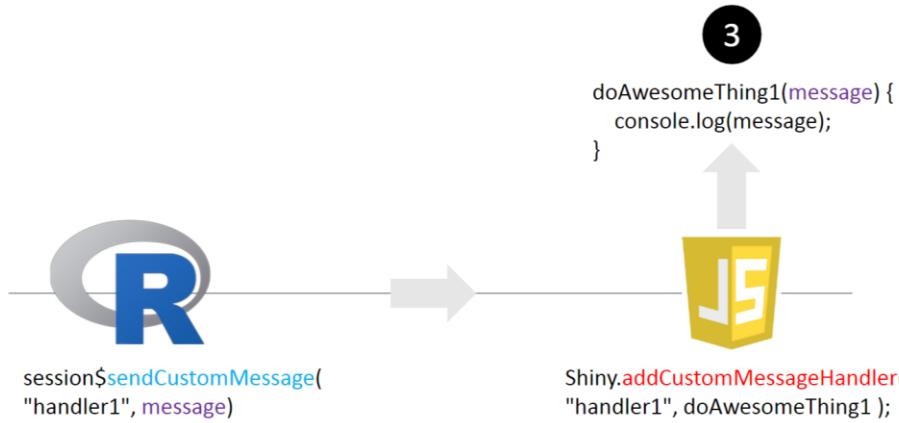
- Introduction
 - Factory methods and closures
- Adding transitions to a C3 gauge
 - Code repository and demo app
 - Htmlwidget fiddle app
 - About the app
- C3 data update pattern
- C3 pie chart
- C3 combination chart
- C3 stacked area chart
- Next tutorial

Introduction

<http://shiny.rstudio.com/tutorial/>

Scenario 1: calling a JavaScript function from shiny

Many modern R packages like `leaflet`, the `DT` package, `visNetwork` and `plotly`, use Javascript functions behind the scenes to perform key functionality. However, as they are all R packages, they invoke such functions from R. How can we do that? Consider the following figure.



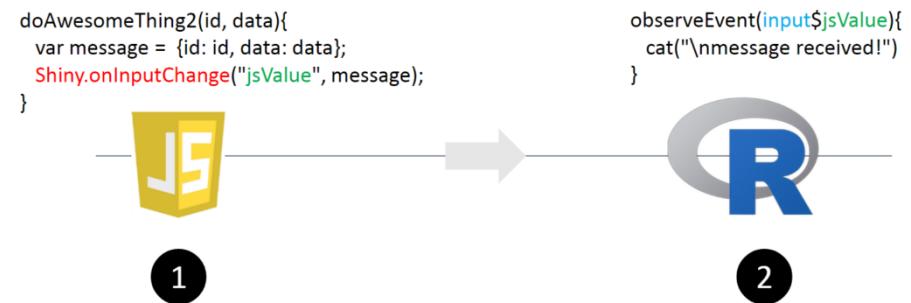
Intro.js basics

Using intro.js in a piece of HTML code is very simple. Let's set up a jsFiddle example using `Bootstrap`, with a grid of 3 rows, with 3 columns each. In addition, let's add a button that on click activates introjs.

A screenshot of a jsFiddle page. At the top, there are tabs for Result, HTML, JavaScript, and CSS. Below the tabs, there is an 'Edit in JSFiddle' button and a cloud icon. The main area contains three rows of three light-grey rectangular boxes each, labeled 'element 1' through 'element 9'. At the bottom center is a green button labeled 'start'.

Scenario 2: sending a message from JavaScript to shiny

Instead of sending a message from shiny to JavaScript, we can also send messages from JavaScript to shiny. These actions are often coupled to events e.g. when we click on an element we subsequently want R to do something. Such messages can be sent using the JavaScript method `Shiny.onInputChange`, which is made available by shiny. Consider the figure below.



In this example we have a JavaScript function `doAwesomeThing2`, which takes an id and some data, who's task it is to send this information to Shiny. In the function we first create an object with name `message`, and subsequently use it to send this message back to shiny. Here we tell it to make it available under the name `jsValue`. Next, in R we can now listen for events via `input$jsValue`. So now if `doAwesomeThing2` is called, shiny gets a signal. Nice!

<http://shiny.rstudio.com/tutorial/>

This repository Search Pull requests Issues Gist

Unwatch 14 Unstar 25 Fork 12

Code Issues 0 Pull requests 1 Wiki Pulse Graphs Settings

tutorials materials on R/Shiny bindings to popular javascript libraries — Edit

46 commits 1 branch 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

		Latest commit 86c3eb5 13 hours ago
	FrissESC added click handlers for some charts	
	updated hyperlinks	2 months ago
	initial commit	4 months ago
	added click handlers for some charts	13 hours ago
	Removed Jquery dependancy from the C3 package such that there is no n...	4 months ago
	initial commit	4 months ago
	version files	4 months ago
	Adds Garrett's .Rproj to .gitignore	4 months ago
	initial commit	4 months ago

<https://github.com/FrissAnalytics/shinyJsTutorials>

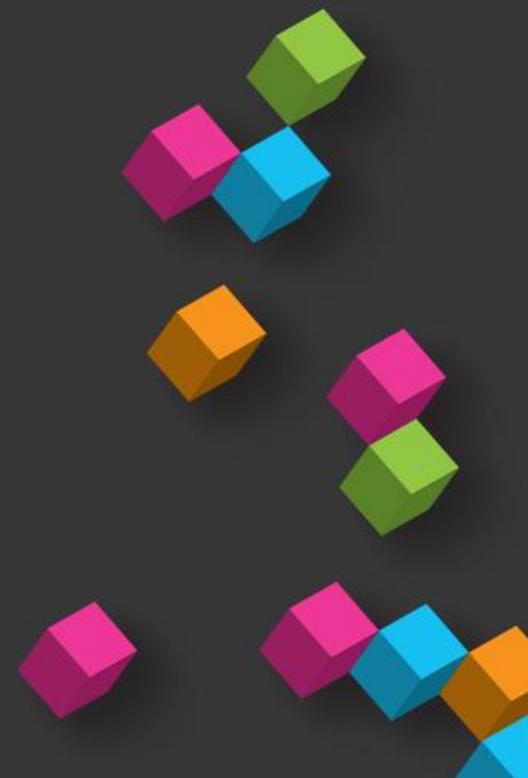
thank you!





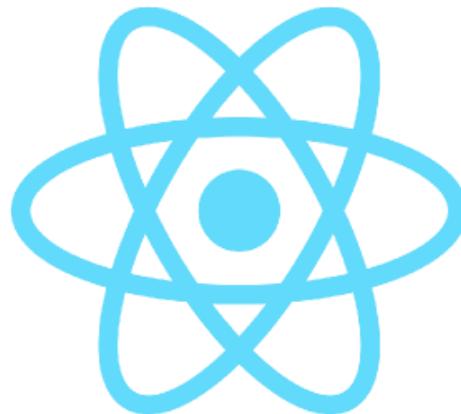
F R I S S
fraud, risk & compliance

Do I need shiny? What else?
alternative tools?





aurelia



React



ANGULARJS²
by Google