Caché User Group (CUG)

Meeting 24/11/2015
Crowne Plaza Antwerp

Agenda

- Evaluation of REST/JSON Workshops, future webinars/workshops
- Presentation of a web application using CSP, Caché SQL & Bootstrap
  - By Christophe Drieghe (CRC)
- Open discussion with all CUG members about development

CUG Workshops

- REST/JSON in Caché with AngularJS
    - 40 participants
    - 3 days (2x Vilvoorde, 1x Amsterdam)
- EWD.js + Node.js (mobile) applications
    - On-line course
- Zen Mojo: TBD*
- CSP/HTML5/Ajax tips & tricks with JQuery/JQuery Mobile
    - Half day seminar in Amsterdam (Simar) - date & mail will follow
- Other tips of (on-line) courses & training ? e.g. Security, (web)servers, ...?

Further details will be posted on the CUG blog:

http://cug-benelux.be

EWD.js / Node.js training

- Online courses on Udemy (step-by-step, starting from scratch):
  - Getting started with EWD.js : http://bit.ly/1lwz5uT
  - EWD.js with Global-based databases: http://bit.ly/1j9nIXU

⚠ CUG members: discount code (valid until 15/12/2015)

- Look at the PoC app with AngularJS: http://bit.ly/1kHsWf4
- Watch EWD.js in action on YouTube!
  - Using React Native: https://www.youtube.com/watch?v=zOIwTU6oPuo
  - Using Bootstrap: https://www.youtube.com/watch?v=I7LMFzkN5_s
- And … why you should embrace JavaScript: http://bit.ly/1OSsEyJ

10 good reasons to start using JavaScript with Caché (connector + EWD.js + Node.js)

1. JavaScript (Node.js) is growing very rapidly and (big) companies are embracing it for enterprise development; Node,js is hugely scalable and runs on commodity hardware (one of the biggest benefits of Caché too!)

2. EWD.js allows you to write (mobile/web) applications with any JS (native) framework of your choice, but write your application code independently

3. EWD,js is very lightweight, doesn't sit "in the way". Gives you a secure connection to the server and hides all the "plumbing", you can concentrate on your application code.

4. EWD,js brings the asynchronous nature of JS nicely together with your synchronous (COS) code you're used to

5. Scalability and resilience: EWD.js uses multiple child processes in the back-end (auto-reconnect, error trapping, load balancing, ...)

10 good reasons to start using JavaScript with Caché (connector + EWD.js + Node.js)

6. EWD.js is open-source and community-driven: if you should want to, you're in complete control (source code is small, maintainable and easy to understand)

7. Multi-mode: using WebSockets, Ajax based or work via the REST (federation!) server; easily customise the way it calls your code; enhancements & new features are continuously been added, your code stays backward-compatible!

8. Gives you the benefits of the Node.js environment with 200.000+ of ready-to-use modules, no need to re-invent the wheel! http://www.modulecounts.com/

9. You can hire developers who know JavaScript and let them work with/getting used to the Caché world … Node.js is the "magic bridge" you need (you can easily use Caché classes and SQL using small wrapper functions)

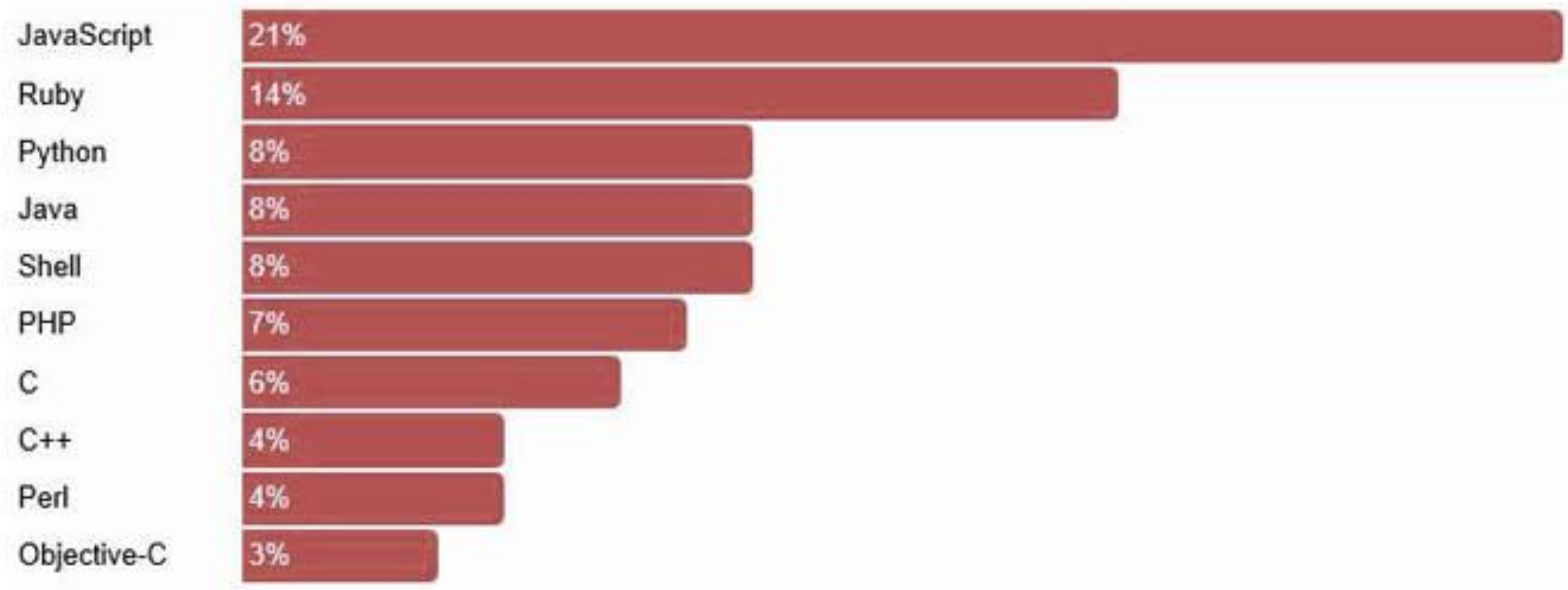
10. And last but not least: these kinds of (mobile) applications give you device and platform independence your customers are asking for!

JS frameworks supported “ootb” by EWD.js

- AngularJS, see: http://bit.ly/1kHsWf4
- jQuery, see: http://bit.ly/1Id8SpQ
- Bootstrap: best example is ewdMonitor itself! ☺
- React,js, see: http://bit.ly/1YreLcl
- React Native, see: http://bit.ly/1jhBIyM
- GraphQL & Relay, see: http://bit.ly/1MzzFkc
- Sencha ExtJS, see http://bit.ly/1Id8SpQ (or contact the CUG)
- ... use any framework you like ...

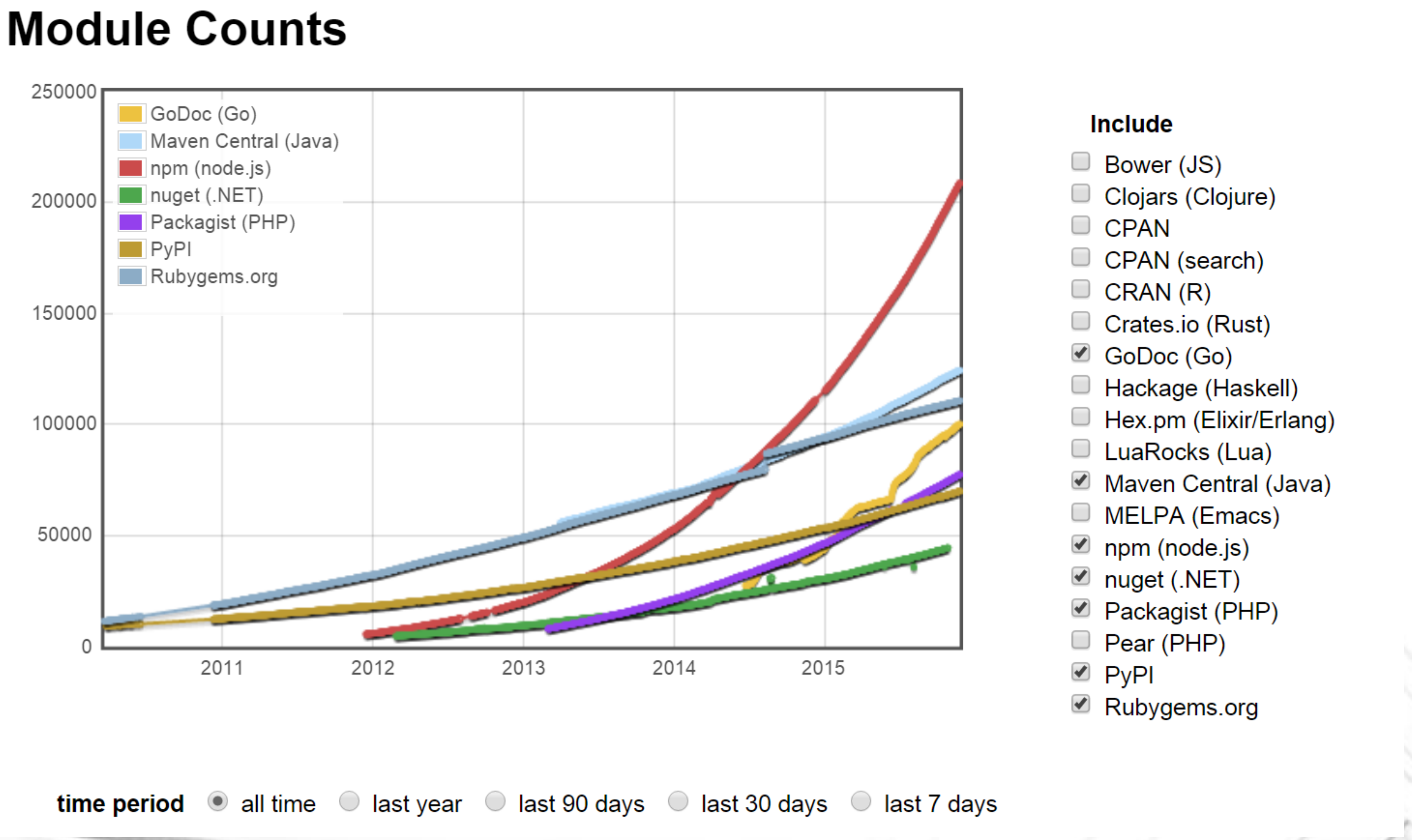
Follow the EWD.js Google group at http://bit.ly/ewdjs-group

JavaScript’s popularity on GitHub



Not convinced yet? Read this … http://bit.ly/1OSsEyJ

Node.js module counts



Module Counts

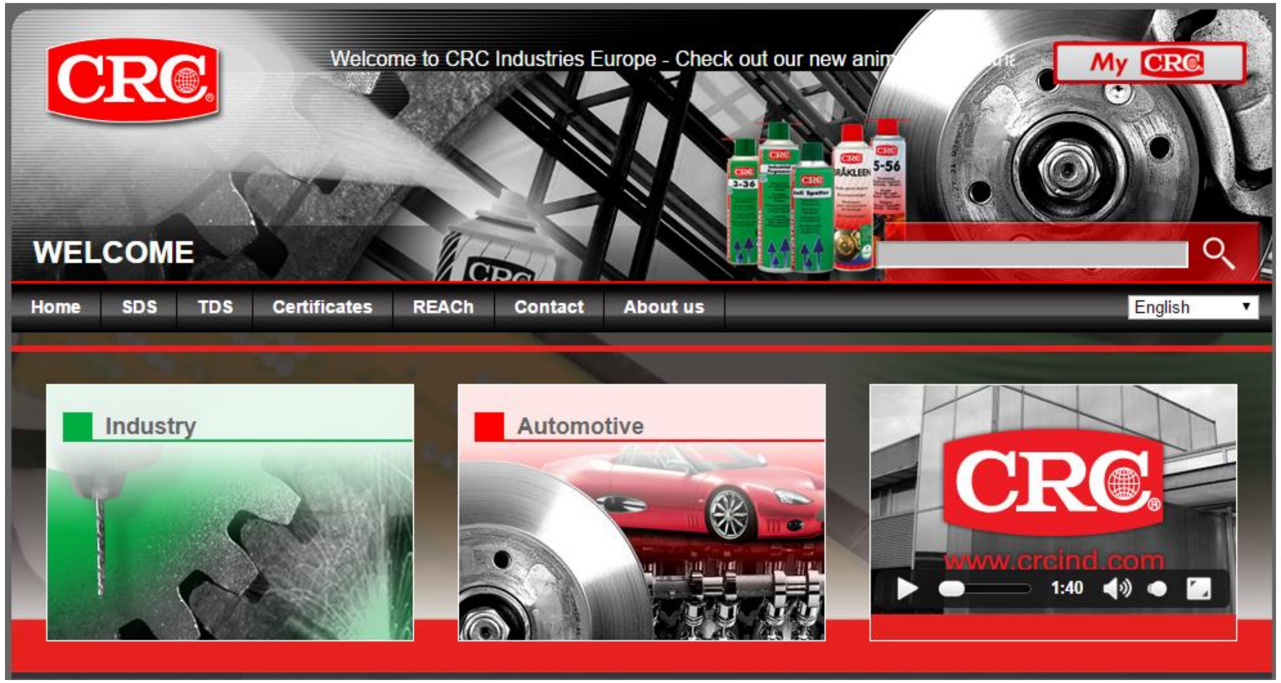
Include

- ☐ Bower (JS)
- ☐ Clojars (Clojure)
- ☐ CPAN
- ☐ CPAN (search)
- ☐ CRAN (R)
- ☐ Crates.io (Rust)
- ☑ GoDoc (Go)
- ☐ Hackage (Haskell)
- ☐ Hex.pm (Elixir/Erlang)
- ☐ LuaRocks (Lua)
- ☑ Maven Central (Java)
- ☐ MELPA (Emacs)
- ☑ npm (node.js)
- ☑ nuget (.NET)
- ☑ Packagist (PHP)
- ☐ Pear (PHP)
- ☑ PyPI
- ☑ Rubygems.org

time period ● all time ○ last year ○ last 90 days ○ last 30 days ○ last 7 days

Demo of CSP + Caché SQL + Bootstrap

- Christophe Drieghe



Development in Caché

- Architecture (Where is your code ?)
  - Split code/data across different databases
  - Split code across servers (develop-test-live)
  - Split code across developers (on client pc's vs. on development server)
- Source code (Any copies if something goes wrong ?)
  - Versioning
  - Backup
  - Other files (csp, images, ...)
- Debugging

Questions for the audience

- How many Caché developers in your organisation ? Non-Caché developers ?
- Are they working on the same code base, same projects ?
- Where is your source code (on different servers, developer pc’s, …)
- What is your strategy for keeping the source code unique, updated, secure?
- How do you integrate COS code with 3rd-party code (JS frameworks, css-libraries, dll’s, .NET, …) ?
- What are your (development architecture) plans for the future?

CUG Benelux

Blog: http://cug-benelux.be

E-mail: info@cug-benelux.be

Twitter: @cugbenelux

LinkedIn group (discussions): CUG Benelux

Announcements, presentation slides will be posted on the CUG blog. Please also follow us on Twitter, join the LinkedIn group (easily become a member by joining the group) and feel free to discuss topics online! If you have questions, suggestions for the CUG core members, you can also contact us by e-mail.