Title:

Remote Work: Martians Style

Talk #:

By:

Sergey Dolganov

Description:

Evil Martians is a highly distributed team of experienced engineers, but starting to work remotely is a significant change in your career.
I want to tell several stories, how my colleagues and I started to work remotely after years of office-only jobs.
Agenda:
- What is so special about working remotely?
- How to treat communication issues?
- How to organize your time?
- How to fight the burnout?
- We all heard stories about remote developers who are traveling all the time, is it real? How to keep being productive on the road?

Votes:

Long Lost

Title:

Continous-Delivery in the Ruby-World

Talk #:

By:

Tobias Münch

Description:

Continous-Delivery in the Ruby-World - but how and which tools you should use?

Agenda:
- Basics of Continous-Delivery
- Basics Component Based Software Engineering
- Ruby-World - Components, Gems, Gem-Hosting
- Basics Docker (build, tag, publish, connect)
- Put it all together -> A Concept in the Ruby-World
- Pratical usage by example

Votes:

RUBY CONF

Title:

Building Resilient API Dependency. No Magic Involved

Talk #:

By:

Sergey Dolganov

Description:

Imagine that you want to build a system which depends on external service, e.g., logistics, payments or notifications service. Those systems have its life-cycle which you have to be in sync with. Also, your own system becomes distributed.
I'll share how to treat issues you could face, using the stories of DHL, UPS, Russian Post and eBay integrations.

I want to tell the story of combining different solutions to find the best way of implementing any integration in your application.

A year ago we started to use design by contracts approach for resilient API client implementation. This year we pushed that technique even further using Types and Functional Approach. So I'll introduce pure Ruby library without dependencies, the most straightforward tool for contracts creation with no magic involved.

Another tool we use now is a «sagas» implementation in Ruby, which helps to implement transactions in a distributed system.

So, the plan is to cover the following topics:
— iterations our application passed through, fighting API client issues
— list of most critical problems in communication between services
— how to treat them in Ruby way
— a quick demo of our architecture, how we applied those solutions

Votes:

Title:

Settings

Talk #:

By:

Andreas Finger

Description:

A clean way to handle **custom configuration values**

in any Ruby app (not only Rails)

without adding a new dependency

and without using `method-missing`.

I will explain the motivation why we started using it (and why I am now using it in every Ruby project).

And will show you a few code snippets (which means all the code there is, as it is really small).

So you never have to read values from ENV variables in your Ruby apps ever again!

Alternative title: How to go from Convoluted Custom Configuration Chaos to Consistent Concise Custom Configuration or CCCC2CCCC

Votes:

RubyConf

Title:

Attacking own APIs to find security bugs

Talk #:

By:

Description:

In this presentation I will talk about how I do my job as a penetration tester for mytaxi. Since we have a huge codebase, lots of API calls, it is really difficult to find the starting point for the attacks.

The best way I found is to try to act as an attacker from the outside, who doesn't know how our systems work, and uses the API calls visible from outside to write a CLI tool that will allow to automatize as many tasks as possible.

Using those scripts, I can implement custom attack scenarios that would be difficult to do manually. By using a web proxy, I can also fuzz the interesting parameters and check for unusual responses.

Agenda:
* Root android/Jailbreak iOS
* Install new certificate authority
* Sniff traffic from application
* Implement the API calls with a python script
* Send requests through the web proxy
* Fuzz interesting parameters
* Create attack scenarios using the findings from above

Votes:

LOST CONF

Title:

Compiling ruby to binary

Talk #:

By:

Torsten Rüger

Description:

Ruby is slow, because it is being interpreted. THEY say that this is because it can not be compiled.

In this talk, i will show that THEY are wrong! Ruby can be compiled, it is "just" a matter of how to get the details right. I will show these details and outline how far i got in the 4 years i have been working on this.

The project is called rubyX (crystal was taken), see ruby-x.org . I can give an overview (not too detailed) including:
- basic compilation idea
- how to bootstrap a compatible ruby
- status of many significant parts that work
- overview of architecture
- overview of calling convention
- overview of binary generation / platform interface

Off course all in 100% ruby, and not even that much / complicated

Votes:

Title:

Zeitwerk

Talk #:

Idea

Interest By:

Jan Krutisch

Description:

Xavier Noria's new loader looks fascinating and I would love to learn more about it from someone who already played with it

Votes:

Lunch Conf

Title:

Webpacker

Talk #:

Idea

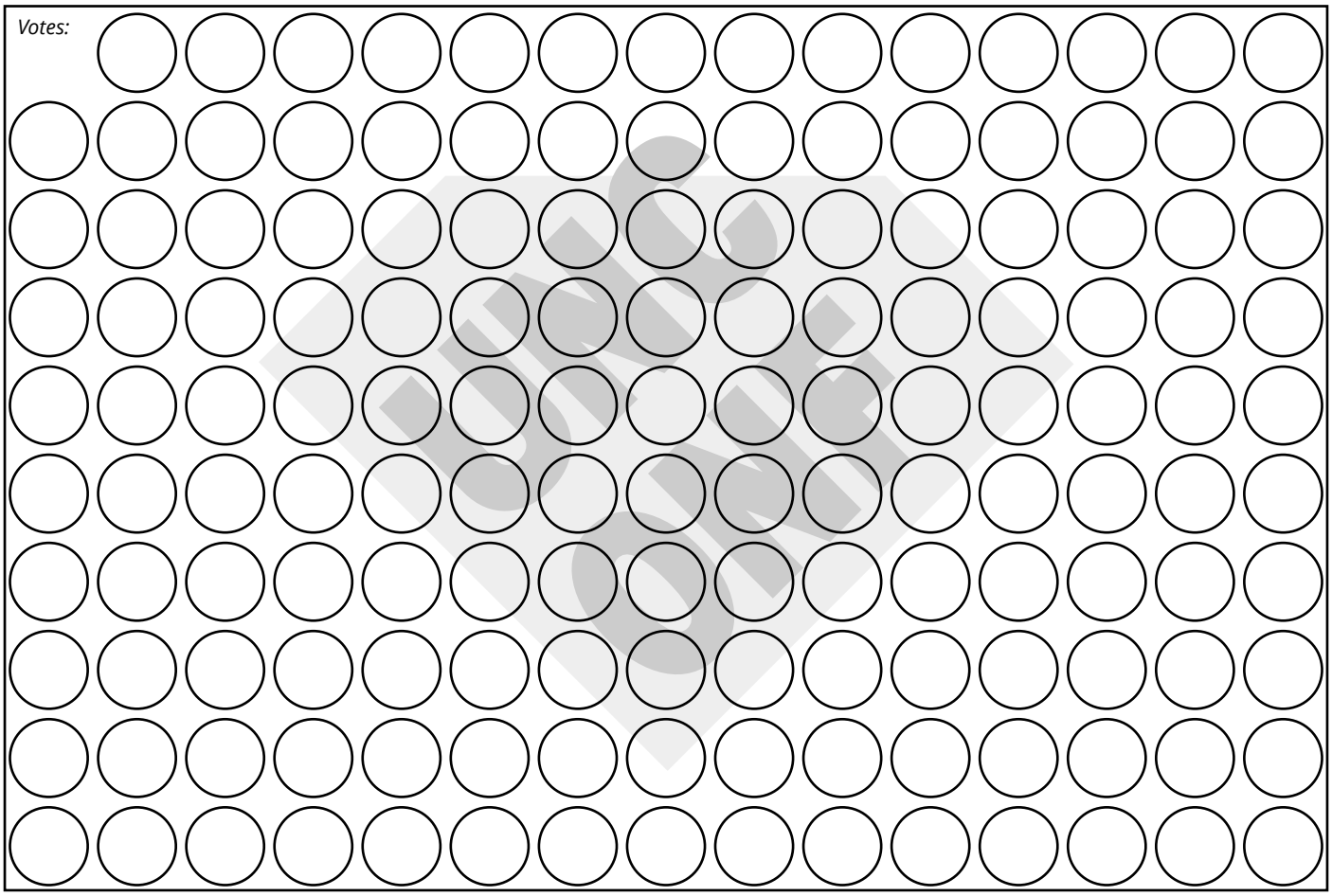
Interest By:

Paul Mölders

Description:

Did you replace the asset pipeline with it? Did you try it? Would love to hear any war stories!

Votes:



Title:

Talk #:

By:

Description:

Votes:

LINUX FEST