

---

08:30 – 09:00	<b>Registration</b> <i>Speakers: Registration</i>	1. Alfa
09:00 – 09:15	<b>Welcome talk</b> <i>Speakers: Welcome talk</i>	1. Alfa
09:00 – 18:00	<b>Open Space</b> <i>Speakers: Open Space</i> <i>Sponsors: 4Finance, Devbridge, Storebrand, Visma Lietuva, WIX Lietuva</i>	6. Lobby
09:15 – 10:15	<b>B Uncle Bob / Robert Martin @unclebobmartin - The Last Programming Language</b> <i>Speakers: Uncle Bob / Robert Martin</i> For the last 50 years we've been exploring language after language. Now many of the "new" languages are actually quite old. The latest fad is "functional programming" which got it's roots back in the 50s. Have we come full circle? Have we explored all the different kinds of languages? Is it time for us to finally decide on a single language for all software development? In this talk Uncle Bob walks through some of the history of programming languages, and then prognosticates on the future of languages.	1. Alfa
10:15 – 10:35	<b>Coffee/tea break</b> <i>Speakers: Coffee/tea break</i>	1. Alfa
10:35 – 11:30	<b>I Dmytro Mindra @dmytromindra - Refactoring Legacy Code</b> <i>Speakers: Dmytro Mindra</i> Every programmer has to face legacy code day after day. It might be ugly, it might look scary, it can make a grown man cry. Some will throw it away and try rewriting everything from scratch. Most of them will fail.  Refactoring legacy code is a much better idea. It is not so scary when you take it in very small bites, introduce small changes, add unit tests. When code is refactored and unit tests are added, changes to functionality can be introduced.  We will take an open source c# project and will refactor it showing step-by-step examples of the techniques.  This session is full of tips and tricks you can start applying immediately. Although the code is in C#, the same principles can be applied in any language	5. Theta
10:35 – 11:30	<b>I Gil Tayar @giltayar - Old Gods &amp; New: A Vision of Backend &amp; Frontend</b> <i>Speakers: Gil Tayar</i> What would happen if we gave front-end developers the task of building a backend server that caters to their needs? What would it do? What capabilities would it have? How would it be different from the current backend servers, built by backend developers? I explore the possibilities and try to envision a future where the front-end developers are in charge of the servers that serve their own front-end code.	3. Lambda
10:35 – 11:30	<b>I Jeremy Gibbons @jer_gib - Categories for the Working Programmer</b> <i>Speakers: Jeremy Gibbons</i> The Haskell community is famous - perhaps infamous - for its enthusiasm for category theory. Why is this? Is it important to understand categories before you can understand Haskell programs? Is it an attempt to keep the community as pure as the language? Is it just that Haskell is a refuge for underemployed mathematicians? None of the above!  In this talk, I hope to explain a little bit about how categories can help the working functional programmer. I'll focus on categories as an organising principle, helping us to manage generic libraries. No monads were harmed during the making of this talk.	2. Beta
10:35 – 11:30	<b>I Oren Eini (Ayende Rahien @ayende) - Building blocks of a distributed system</b> <i>Speakers: Oren Eini (Ayende Rahien)</i>	1. Alfa

In this talk, Oren will discuss the building blocks of building a reliable, transactional distributed database. In particular, this session will cover ACID compliance, ensuring consistency between distributed nodes (with failure handling), monitoring and management, dissemination of information in the system, and more.

---

10:35 – 11:30 **I Paul Stack @stack72 - Creating a scalable, repeatable infrastructure with Terraform** 4. Zeta

*Speakers: Paul Stack*

The age old task of racking and stacking in a physical data centre is becoming more and more rare as more companies embrace the public cloud. Having the ability to choose between providers such as AWS, Azure, Digital Ocean and Google Cloud Platform makes creating infrastructure easy. It is better to spend time developing better services for our customers than managing infrastructure

During this talk, Paul will demonstrate how building a scalable infrastructure on AWS becomes easy with Terraform. The talk will demonstrate how using configuration management, pre-baked AMIs and auto-scaling groups it gives the ability for developers to be able to launch their own infrastructure when needed. The demo's will include the ability to launch instances, databases and manage user access

By the end of the talk, Paul will have demonstrated that the creation of infrastructure now becomes part of the development lifecycle and that the old ways of system administration is fast moving to become infrastructure engineering. Paul will also demonstrate that the creation of new 'environments' are just a change of parameters in our infrastructure code

---

11:50 – 12:45 **B Nakul Mishra - Microservice - no fluff the REAL stuff** 5. Theta

*Speakers: Nakul Mishra*

Some developers tend to believe that big data and huge traffics are pre-requisites to harness and reap the benefits offered by microservices. However, there are many problems like polyglot persistence, faster continuous build cycle, shorter release plan, etc. that can be solved by applying microservice architecture even in smaller organizations. You don't have to be a software giant to leverage stability, effectiveness and flexibility provided by microservices. Microservice architecture has a lot to offer even for smaller organizations. In this talk, we will walk through the key concepts like service discovery & registry, circuit breaker, API gateway and edge service. Build a bunch of microservices demonstrating practical implementation of these patterns using open source components like Eureka, Zuul, Hystrix and powered by spring cloud. Lastly, we will deploy our microservices on a container based solution, i.e. Docker and discuss how tools like Chaos Monkey and Janitor Monkey can help making our application fault-tolerance and keeping the cloud's infrastructure neat and tidy.

---

11:50 – 12:45 **B Pieter Hintjens @hintjens - Building Open Source Communities** 4. Zeta

*Speakers: Pieter Hintjens*

Whether you make open source or use it, one thing is clear. Without community, an open source project will fail. In this talk Pieter boils 30 years of experience down to ten rules for building a successful, happy open source community. Rule number 1 is "People before code." He explains this rule, and the other nine rules, with examples from the ZeroMQ community and other projects.

---

11:50 – 12:45 **I Ali Kheyrollahi @aliostad - 5 Anti-Patterns in designing APIs** 3. Lambda

*Speakers: Ali Kheyrollahi*

This talks elaborates on the Client-Server tenet of REST which focuses on separation of concerns between the client and the server. In the first third of the talk, I will talk about what the ideal client and servers are and examples of how their responsibilities. I will touch on how the word Server has lost its meaning of "serving" and the client has been overshadowed by the focus to the API. I will also compare the API to a restaurant and how its menu is the API's REST resources.

In the rest of the talk, I look at some important anti-patterns commonly seen in the industry (each with at least one example):

1) Chauvinist Server: designing the API from server's perspective failing to hide its complexity behind its API (API

designed from the server's perspective)

2) Demanding client: client enforcing its special need onto the signature of the API (certain client's limitation becomes server's default behaviour)

3) Transparent Server: server exposing its internal implementation to its clients (server's underlying or private domain bleeds into the public API)

4) Presumptuous Client: The client assuming the role of a server and engage in taking responsibilities that cannot guarantee

5) Assuming Server: Server that assumes the responsibility of tailoring the response based on what it assumes client is (e.g. browser sniffing)

---

11:50 – 12:45 **I Aurimas Adomavicius @needoptic - Great User Experience Through Dual-Track Scrum** 2. Beta

*Speakers: Aurimas Adomavicius*

*Sponsors: Devbridge*

This talk builds on top of the content presented at the closing keynote of Agile Tour Vilnius 2014 - "Using agile with fixed bid projects". A lot of companies struggle weaving design and development throughout their implementation of Agile. We would like to share our story of adopting agile and transitioning into a dual-track model for lean discovery, User Experience Design, and implementation. Core structure of the presentation:

- Metrics of great User Experience for the enterprise
- Dual-track - model, common pitfalls, lessons learned
- Closing the loop and using user metrics to quantify success of project (User Testing, etc.)

More information is available upon request.

---

11:50 – 12:45 **I Rob Ashton @robashon - The Shape an Erlang Application** 1. Alfa

*Speakers: Rob Ashton*

Enough introductions to Erlang, let's ignore the language for a moment and have a look at deeper things - how do you build and release Erlang projects? How do you structure Erlang projects? What are some common pitfalls to avoid when putting together Erlang applications, how do you manage dependencies?

Let's talk about the real stuff based on the last two years of my working as a full-time Erlang developer, lots of code and examples in a whirlwind tour - do try to keep up.

---

12:45 – 13:45 **Lunch** 1. Alfa

*Speakers: Lunch*

---

13:45 – 14:40 **B Jef Claes @JefClaes - Evil by Design** 5. Theta

*Speakers: Jef Claes*

In this talk, I'll share what my experience has been working in the gambling business, how moving to events helped us gain a better understanding of the domain and which techniques and models casinos have perfected over the years to keep people playing.

---

Last year I ventured into the domain of (online) gambling. Given that the industry has been around since forever, I expected most problems to be of the technical kind. As it turned out, the struggle with technology was only part of a bigger problem; to move forward we needed to fully grasp the industry and its consumers.

Domain events started out as a way to dismantle a legacy system, but quickly proved to be an effective tool to gain a deeper understanding of our domain. Visualizing event streams, we discovered patterns that helped us identify what drives different types of users.

Having a better understanding of what customers are looking for, we dove into existing literature to learn which techniques and models casinos use to cater for each type of user. We learned how to program chance while staying true to the Random Number God. Even when variance is brutal, casinos have enough data and tools to steer clear from the pain barrier.

All of this entails interesting problems and software, but isn't my code damaging society? Or is gambling just another human trait?

---

13:45 – 14:40

**I Itamar Syn-Hershko @synhershko - Logging makes perfect - real-world monitoring and visualizations**

4. Zeta

*Speakers: Itamar Syn-Hershko*

How to keep a real-time, low-latency and high-stakes system up and running and well-monitored? how to investigate failure cases as they happen? and how to even know something is wrong before it's too late? With logs of course. Lots of them. And then some cool stack to do stuff with it.

Forter is a company with a Decision-as-a-Service product that deals with many e-commerce transactions in real time and answers a simple but hard question: "is this a fraud attempt or not?". And if we were wrong, we pay.

In this talk I will show how we are using various technologies to power our service and keep it high-available and well under control. Among the technologies I will discuss are Apache Storm, Node.js, Riemann (state machines in Clojure, yay!), collectd, D3.js and of course the ELK stack (and beats!). Some integrations which will be mentioned include PagerDuty, Slack, Jenkins and GitHub.

---

13:45 – 14:40

**I Reynhout Yves @bittacklr - Trench Talk: Models and friends**

3. Lambda

*Speakers: Reynhout Yves*

This talk takes the janitor's cut to models and friends. How they're crunched, born, tested against scenarios, how they're useful, what distinguishes them from others, how they're visualized and communicated, how they change over time, how they do not always turn out the way you want them to be, how some of them turn to mud, how they don't life in isolation, how ... well, you'll just have to attend to hear more, won't you?

---

13:45 – 14:40

**I Richard Minerich @rickasaurus - How We Use Functional Programming to Find the Bad Guys**

2. Beta

*Speakers: Richard Minerich*

Traditional approaches in anti-money laundering involve simple matching algorithms and a lot of human review. However, in recent years this approach has proven to not scale well with the ever increasingly strict regulatory environment. We at Bayard Rock have had much success at applying fancier approaches, including some machine learning, to this problem. In this talk I walk you through the general problem domain and talk about some of the algorithms we use. I'll also dip into why and how we leverage typed functional programming for rapid iteration with a small team in order to out-innovate our competitors.

---

13:45 – 14:40

**I Venkat Subramaniam @venkat\_s - Rediscovering JavaScript**

1. Alfa

*Speakers: Venkat Subramaniam*

JavaScript is one of those very powerful languages that is often misunderstood and underutilized. It is quite popular, yet there is so much more we can do with it. In this presentation we will deep dive into the capabilities and strengths of this prominent language of the web.

---

15:00 – 15:55

**A Diego Ongaro @ongardie - The Raft Consensus Algorithm**

3. Lambda

*Speakers: Diego Ongaro*

Consensus is fundamental to building fault-tolerant systems, but it's poorly understood. We struggled to build a complete system using Paxos, so we developed the Raft consensus algorithm to be easier to understand. Since releasing our first paper draft in 2012, Raft has been implemented in dozens of libraries and systems, and it's now taught at over ten universities. In this talk, I'll give an overview of how Raft works. More info on Raft can be found at <https://raft.github.io>.

---

15:00 – 15:55

**B David Larabee @larabee - The Liberal Arts Programmer**

1. Alfa

*Speakers: David Larabee*

At a certain point software was everything in my life. Learning new languages, architectures, design patterns and acquiring skill in practice - my raisons d'etre. After a while things clicked, and I started leading teams of more junior programmers.

"The People Problem" presented new and foreign challenges. There's nothing quite like working with others on real products to transform egocentric practice into empathic pragmatism. My search for a new bag of tricks led me back to my earlier studies in the humanities. Suddenly my liberal arts education, formerly regarded as tangential even if interesting, felt applicable to more than cocktail party conversation.

In this talk we'll look at disciplines outside the purview of software development with an eye for how borrowed ideas can yield not only inspiration and analogy, but real innovations and breakthroughs. I'll share examples from

fine art, psychology, history and literature influence my approach and the work of others.

My sincere hope is that you'll walk away from our time together with a heightened appreciation for how looking outside software development can enhance your journey within software development.

- 
- 15:00 – 15:55 **B Sean Trelford @ptrelford - Fun and games with F#** 5. Theta  
*Speakers: Sean Trelford*  
In this live coding session I'll show you how to make music, 3D scenes through to interactive video games with F# in the REPL.
- 
- 15:00 – 15:55 **B Yan Cui @theburningmonk - Seven ineffective coding habits many F# programmers don't have** 4. Zeta  
*Speakers: Yan Cui*  
At BuildStuff'14, Kevlin Henney presented an excellent talk titled "Seven ineffective coding habits of many programmers". As an attendee that day and someone who has exhibited many of these habits over the years, I came to realize that using F# has cured me of many of these ineffective habits!  
  
In this talk I'll share my thoughts on how the use of F# and functional programming techniques can help form and nurture good habits and give you the perfect practice you need to make perfect.
- 
- 15:00 – 15:55 **I Ian Cooper @ICooper - Service Discovery and Clustering for .NET developers** 2. Beta  
*Speakers: Ian Cooper*  
Building a distributed system means you need to consider how you will discover services, and ensure they are available. In this presentation we look at Service Discovery and Clustering approaches and tools, and show .NET developers how to work with tools such as Serf, Consul, and Zookeeper
- 
- 15:55 – 16:15 **Coffee/tea break** 1. Alfa  
*Speakers: Coffee/tea break*
- 
- 16:15 – 17:10 **A Howard Chu @hyc\_symas - The Lightning Memory-Mapped Database** 2. Beta  
*Speakers: Howard Chu*  
The Lightning Memory-Mapped Database (LMDB) was introduced at LDAPCon 2011 and has been enjoying tremendous success in the intervening time. LMDB was written for the OpenLDAP Project and has proved to be the world's smallest, fastest, and most reliable transactional embedded data store. It has cemented OpenLDAP's position as world's fastest directory server, and its adoption outside the OpenLDAP Project continues to grow, with a wide range of applications including big data services, crypto-currencies, machine learning, and many others.  
  
The talk will cover highlights of the LMDB design as well as the impact of LMDB on other projects.
- 
- 16:15 – 17:10 **B Greg Young @gregyoung - Privateeye** 1. Alfa  
*Speakers: Greg Young*  
In this talk we will sleuth into what is privateeye. We will turn our detective skills on how your application actually work and we will do it using nothing but a REPL. You know how to code, let's code through a murder mystery together.
- 
- 16:15 – 17:10 **B Jevgenij Nekrasov @jnekrasov - Being Meta** 5. Theta  
*Speakers: Jevgenij Nekrasov*  
*Sponsors: Visma Lietuva*  
We are going to discuss meta programming approaches in .NET world, trying to give brief overview of different techniques, which can expand your horizons as a developer and become more meta-developer. It's all about writing code, which analyse your code or manipulate it, so we will start with some simple examples and then go a bit dipper. Special attention will be given to DSL and how you can construct your own implementation.
- 
- 16:15 – 17:10 **I Robert Virding @rvirding - Implementing Languages in Erlang** 3. Lambda  
*Speakers: Robert Virding*  
This tutorial will look at the problem of implementing languages in Erlang on top of the Erlang system. Such

languages can be anything from small DSL for coding a specific problem to a complete language which for some reason it is beneficial to run it natively inside Erlang. We will look at parsing languages, implementing a basic interpreter and then integrating that into an Erlang application. Finally we will discuss compiling our languages into Erlang. As an example we will use a small existing language. The topics we will look at here are quite common and apply to most languages and many of the solutions are also applicable to other languages and systems apart from Erlang.

---

16:15 – 17:10 **I Sebastien Lambla @serialseb - The Simple Life of ReSTful Microservices** 4. Zeta

*Speakers: Sebastien Lambla*

Microservices are in, monolithic apps are out, everyone is high in the cloud, SOA is undead, ReSTful is never really ReSTful because pragmatism, and messaging is either really awesome or really bad. Very complicated, right? Complex systems are all around us, often made of many small and simple entities. In this talk we'll explore how complexity can be reduced to its smallest cohesive parts, communication normalized through the power of evolvable contracts, ReSTful and event-driven interfaces, and how to make a microservice swarm fly in unison.

---

17:10 – 18:35 **B 'Beer time' with Visma brewed beers! :)** 1. Alfa

---

## NOVEMBER 19 • THURSDAY

---

08:45 – 09:10 **Morning coffee/tea** 1. Alfa

*Speakers: Morning coffee/tea*

---

09:00 – 17:05 **Open Space** 6. Lobby

*Speakers: Open Space*

*Sponsors: 4Finance, Devbridge, Storebrand, Visma Lietuva, WIX Lietuva*

---

09:10 – 10:10 **B KEYNOTE: Mel Conway @conways\_law - Coding vs. the Brain: Can't We All Just Get Along?** 1. Alfa

*Speakers: Melvin Conway*

In an extremely short time interactive information appliances such as mobile devices, computers, and interactive kiosks such as ATMs have exploded into common use all over the globe. An understanding of how these appliances work must now join arithmetic and the calendar in the migration toward universally accessible simplicity. This migration will require a radical simplification of the conceptual model for the internal workings of interactive appliances that is more intuitive than algorithms for the mass of people.

The talk presents a hybrid unidirectional-flow/message model of the internal operation of interactive information appliances that is intuitive, generally applicable, and largely algorithm-free. It also presents design principles that formalize what-you-see-is-what-you-get construction-tool behavior. Finally, the talk demonstrates a proof-of-concept application builder that conforms to these design principles and that builds small applications that work according to the new conceptual model.

---

10:10 – 10:30 **Coffee/tea break** 1. Alfa

*Speakers: Coffee/tea break*

---

10:30 – 11:25 **B Simona Bekeraite @technarium - Building Stuff at Technarium: a hackerspace in Vilnius** 5. Theta

*Speakers: Simona Bekeraite*

Technarium is an independent, community-operated hackerspace in Vilnius, Lithuania. We make things: analogue and digital physical stuff, software, research experiments, art. In this talk we'll show some of our projects, talk about the stuff that keeps us ticking and tell about the joys and difficulties of building an independent open space of technical creativity.

---

10:30 – 11:25 **I Felienne Hermans @Felienne - A board game night with geeks** 2. Beta

*Speakers: Felienne Hermans*

So this one day, I am playing the board game Quarto ([http://en.wikipedia.org/wiki/Quarto\\_%28board\\_game%29](http://en.wikipedia.org/wiki/Quarto_%28board_game%29)) with my friend and I wonder, can this game end in a tie, or is there always a winner?

Normal people might have squabbled or shrugged, but not us nerds! We obviously abandoned the game, took our laptops to the local pub and started hacking. In this talk I will explain how I used F# to transform this problem to satisfiability, and ran it through a sat solver to discover if it can indeed end in a tie.

I will also show how to apply the same technique to more useful problems such as scheduling and register

allocation.

- 
- 10:30 – 11:25 **I Motiejus Jakštys @mo\_kelione - Unikernels and the future of secure cloud computing.** 4. Zeta  
*Speakers: Motiejus Jakštys*  
Over one million AWS customers are happy with the benefits they get from Cloud Computing. One of the reasons for this is the vast array of choice they have in how they run their applications in the cloud. One choice customers have to run their applications, that is not so well known, is to use Unikernels. At the end of the talk, you will understand how Unikernels can make your applications efficient, scalable and secure.  
  
The talk will be followed by a demonstration on how we all can take advantage of unikernels right now. An existing Linux-runnable web service will be converted to a unikernel and executed both the local desktop and on Amazon EC2.
- 
- 10:30 – 11:25 **I Phillip Trelford @ptrelford - Beyond Lists** 3. Lambda  
*Speakers: Phillip Trelford*  
Selecting appropriate data structures is key to your application's performance. In this session we'll go beyond lists to find orders of magnitude performance improvements. Expect plenty of live demos and anecdotes gathered over decades of financial and AAA video game development
- 
- 10:30 – 11:25 **I Venkat Subramaniam @venkat\_s - Transforming Your Code to Java 8** 1. Alfa  
*Speakers: Venkat Subramaniam*  
The new facilities in Java 8 is about the change the way we write code. Our code will become more expressive and concise. But, exactly how? In this presentation we will take several common Java code examples, discuss the core idea expressed in code, and transform that code to use the facilities in Java 8. Watch and interact as you see Java code go through a weight loss program right in front of your eyes.
- 
- 11:45 – 12:40 **B Mathias Brandewinder @brandewinder - The T in TDD: Test, Types, Tales** 4. Zeta  
*Speakers: Mathias Brandewinder*  
Test-Driven Development is about writing a test first, then the code that satisfies the test. Or... is it? Classic TDD has been crucial in my growth as a developer; and yet, as I started using F# more and C# less, my coding practices have evolved dramatically. Nowadays, I barely test first, but spend my days in the scripting environment. In this talk, I will examine how I write code in F# today, in a style I consider very much inspired by TDD; I will also discuss how exploring two paradigms (C# and F#) forced me to re-examine my beliefs, and evolve a slightly different (and perhaps less language specific) understanding of some of the same core principles.
- 
- 11:45 – 12:40 **B Øystein Kolsrud - Functional Programming for the Object Oriented** 5. Theta  
*Speakers: Øystein Kolsrud*  
Most modern programming languages have taken inspiration from the functional programming paradigm and have implemented features for making functional modelling easier. To be able to fully leverage the power of languages such as C#, it is therefore necessary to have an understanding of functional programming as well as other paradigms such as object orientation. This presentation gives an introduction to Haskell and compares it to the features provided in C# that have been influenced by concepts from the functional world. In particular, it aims to illustrate how a functional approach to solving problems differs from an object oriented approach.
- 
- 11:45 – 12:40 **I Ali Kheyrollahi @aliostad - From Power Chords to Power of Models: Insights from History of Rock Music via Machine Learning** 2. Beta  
*Speakers: Ali Kheyrollahi*  
Who were the most influential bands of Rock history? Which bands could not exist there was no Velvet Underground? How much Shoegazing subgenre is related to the Drone music? If I like AC/DC, what is the  
  
Rock music history was perhaps full of drugs and alcohol but we are sobering up to represent it in terms of (social) networks and find mathematical relationship between artists, trends and subgenres. Full of DataViz and interesting relationships, we will pick up a few common clustering and network analysis algorithms to analyse the publicly available Wiki data. Expect lots of air guitar power chords and virtuoso solos.
- 
- 11:45 – 12:40 **I Liz Keogh @lunivore - Why Building the Right Thing means Building the Thing Right** 3. Lambda

*Speakers: Liz Keogh*

Whenever we do anything new, we make discoveries. From small changes to the UI to disruptive innovation; from learning how our team-mates like their coffee to creating brand warmth in a global enterprise; everything we do involves reacting not just to the problems we discover, but also the opportunities. In this talk we look at why experimentation underpins everything we do in technology, and why being able to move and change the right thing

---

11:45 – 12:40

**I Randy Shoup @randyshoup - Service Architectures at Scale: Lessons from Google and eBay**

*Speakers: Randy Shoup*

1. Alfa

Over time, almost all large, well-known web sites have evolved their architectures from an early monolithic application to a loosely-coupled ecosystem of polyglot microservices. While first-order goals are almost always driven by the needs of scalability and velocity, this evolution also produces second-order effects on the organization as well. This session will discuss modern service architectures at scale, using specific examples from both Google and eBay.

It will cover some interesting -- and perhaps nonintuitive -- lessons learned in building and operating these sites. It continues with some more advanced implications of a microservices architecture, including SLAs, cost-allocation, and vendor-customer relationships within the organization. It concludes by exploring a set of common service anti-patterns.

---

12:40 – 13:40

**Lunch**

1. Alfa

*Speakers: Lunch*

---

13:40 – 14:35

**A Chris Condron @CLCondron - Unsafe at any Speed - Successful high performance low latency systems in C#**

3. Lambda

*Speakers: Chris Condron*

A walk through of key pieces of a working production architecture that performs real-time analytics and visualization on 113 million data points per second on a single desktop class workstation. This was achieved through a combination of message oriented processing and unsafe data structures in key locations.

We will review how we mixed managed code across the majority of the application with unsafe data structures in key algorithmic location giving the best of both world.

We will review the details of simple custom memory management used in the allocation unsafe data without leaks or GC thrashing and some of the particulars of the general algorithmic approaches leveraging data locality and pointer operations.

Finally we will review the message based data processing pipeline that routes the processing through the system.

---

13:40 – 14:35

**B Kevlin Henney @KevlinHenney - Functional Programming You Already Know**

1. Alfa

*Speakers: Kevlin Henney*

From JVM to .NET languages, from minor coding idioms to system-level architectures, functional programming is enjoying a long overdue surge in interest. Functional programming is certainly not a new idea and, although not apparently as mainstream as object-oriented and procedural programming, many of its concepts are also more familiar than many programmers believe. This talk examines functional and declarative programming styles from the point of view of coding patterns, little languages and programming techniques already familiar to many programmers.

---

13:40 – 14:35

**B Pavlo Baron @pavlobaron - Why we do tech the way we do tech now?**

5. Theta

*Speakers: Pavlo Baron*

The pace with which we introduce, replace, remove, reinvent, copy, modify and fork technologies has become insane. Even 10 years ago, a developer was focusing on one language, one framework, one database, one area. Today, we eventually have to write code in multiple languages on one single project, mixing multiple databases and going through the whole technology stack of the modern IT. There isn't even time to hold on and ask yourself: why are we doing tech the way we're doing tech today? I'll explain why, and eventually help turning from passive passenger into a co-driver.

---

13:40 – 14:35

**I Dylan Beattie @dylanbeattie - Domain Architecture Isomorphism: How Spotlight Inverted Conway's Law**

2. Beta

*Speakers: Dylan Beattie*

Conway's Law says that "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations". We've seen this happen time and time again.



Some organisations are fortunate enough that their communication structure happens to match their ideal architecture. Some organisations succeed in spite of their communication structures, but all too often the communication overheads result in systems that are delivered late, over budget and unfit for purpose.

But what if we're looking at it backwards? Maybe we should start with the system design, and then create an organisational structures that reflects it?

Dylan is the systems architect at Spotlight ([www.spotlight.com](http://www.spotlight.com)), the UK's leading casting service. Since 1927, Spotlight has been used to cast productions from Monty Python and James Bond to Star Wars and Game of Thrones. Originally a printed directory, Spotlight has been on the web since 1996, and was one of the first companies in the UK to embrace digital publishing. Having successfully made the transition from a paper-based directory publisher to a digital services company, we now find ourselves in the unusual position of being nearly a century old, yet facing many of the same problems as a successful startup. In this session, we'll talk about how we're restructuring Spotlight so that our systems and our structure align with the natural domain boundaries of our business. We'll talk about the challenges we've faced - how do you promote microservices and modular architecture when some of your customers still think fax machines are a pretty neat idea? How can you decouple monolithic legacy systems, identify bottlenecks, and create just the right amount of architecture? We'll talk about systems integration patterns, and how to run the New Shiny alongside your legacy systems to avoid the risk of "big bang" launches. And you never know - we might even throw in a couple of good showbusiness stories.

---

13:40 – 14:35 **N Pete Smith @beyond\_code - The Myth of the Qualified Developer** 4. Zeta

*Speakers: Pete Smith*

There's a lot to master in any job, but software engineering takes this to a whole new level. A good developer has to enjoy learning, and of course we usually do this by making mistakes. But what level of knowledge divides a master from a mere competent beginner? How do we know when we've learned enough to do our job properly and consider ourselves fully qualified?

In this talk I'll help you to answer these questions by sharing the stories of my own greatest mistakes, and reveal how a lot of them ended up becoming my greatest opportunities. We'll explore what it means to fail (sometimes specatacularly) and most importantly, how to make the most of it. And remember - whether you think you know too much, or too little - you're wrong!

---

14:55 – 15:50 **B Dmytro Mindra @dmytromindra - Let's Build a 2D Game!** 5. Theta

*Speakers: Dmytro Mindra*

Dmytro Mindra, just left Unity Technologies, the company that ships one of the best cross platform game engines. He still can teach you some game development if asked ;)

Dmytro will make an introduction to game development in Unity and will show how to make a simple 2D game in just an hour. The material for this talk is simple enough for those who have no experience in working with Unity and will feature some really basic C#.

What will you learn? Attendees will get all the material and knowledge to create a simple 2D space shooter game (vertical scroller).

What do we need? Good mood. Basic programming skills (or at least basic copy and paste skills). Laptop with Unity 5.2 installed, if you want to follow some steps.

Who may come? Everyone, who wants to have fun and to learn how to make a simple 2D space shooter game. And we will give special prizes to those of you who will make the best BuildStuff themed game ;)

---

14:55 – 15:50 **B Mark Rendle @markrendle - ASP.NET 5 on Docker** 1. Alfa

*Speakers: Mark Rendle*

Now that ASP.NET is fully supported on Linux, you can package and deploy your MVC 6 applications using Docker. In this talk, I'll show you: how to use Docker with ASP.NET 5; how to deploy Docker-packaged solutions to cloud or private platforms; and a variety of Docker-related tools that help in development, testing and production.

---

14:55 – 15:50 **B Tomer Gabel @tomerg - Onboarding at Scale: An Engineering Problem** 4. Zeta

*Speakers: Tomer Gabel*

Of the myriad challenges in scaling up an engineering organization, onboarding new employees is probably the least well-understood. There are relatively common solutions for large-scale recruitment, finance and administration, but onboarding remains a question that many organizations struggle with.

At Wix we've been struggling with massive scaling challenges: over the last two years our company headcount has doubled itself, and we had to learn to cope with the influx while maintaining velocity. In this talk we'll share with you the story of how we set up Wix Academy, an engineer-driven training organization, the solutions we've developed (and still are!), and what we've learned in our first year of operation.

---

14:55 – 15:50 **I Ian Cooper @ICooper - Brighter for robust, scalable .NET apps** 3. Lambda

*Speakers: Ian Cooper*

Brighter <http://iancooper.github.io/Paramore/Brighter.html> is an OSS library for .NET that allows you to build a robust, fault-tolerant Command architecture for .NET. It supports pipelined execution of commands both in-process and via a Task Queue, and was highlighted in the May 2015 ThoughtWorks Technology Radar. In this presentation we look at what Brighter is, how it does it, and show you how to build robust distributed system with it.

---

14:55 – 15:50 **I Kristjan Korjus @kristjankorjus - Artificial Intelligence that plays Atari video games: How did DeepMind do it?** 2. Beta

*Speakers: Kristjan Korjus*

We replicated the work of DeepMind as an open source project and created an artificial intelligence that can learn to play different video games in a super-human level without any human intervention. I will talk about deep learning, reinforcement learning and their combination called deep Q-Network.

---

15:50 – 16:10 **Coffee/tea break** 1. Alfa

*Speakers: Coffee/tea break*

---

16:10 – 17:05 **B KEYNOTE: Russell Miles @russmiles - Lies, Damn Lies and Consulting Lies - The Path to World Domination through Microservices** 1. Alfa

*Speakers: Russell Miles*

In this epic, life transforming, talk, Chief Principal Senior Consultant Scientist from Global Enterprise Consultancy ThoughtFlixPivot(tm), will expose the industries best kept secrets on what we now know to be the one true way to a successful Microservices project.

(no JEE Monoliths were harmed in the making of this talk, but may be in the delivery)

---

19:00 – 20:00 **B 7PM-1AM Party night - other venue! EXIT club Jasinskio 16A, Vilnius** 1. Alfa

*Moderators: Dylan Beattie, Mark Rendle*

*Speakers: Party night*

This year party will be held at **EXIT club** (Jasinskio 16A, Vilnius). We prepared separate zones for people who wants to chat or code and those who want to dance and go crazy with amazing DEVELOPERS music band "UNDEFINED" from VISMA LIETUVA. All the zones will have it's own bar - so fill free to grab a beer - it's on us!

**Quiz show-"Have I Got BS For You"**

Also, our star guests will demonstrate their knowledge of the latest news in technology, software development and buzzwords a new topical quiz show. With over 0b1000 rounds of questions, answers, caption competitions, missing words, the Worst Language In The World, Bot-or-Not, Jargon Jeopardy! and more - all keeping within the Code of Conduct, but only just.

Join hosts Mark Rendle and Dylan Beattie bringing together the best of technology, trivia and comedy.

---

## NOVEMBER 20 • FRIDAY

---

09:40 – 10:00 **Morning coffee/tea** 1. Alfa

*Speakers: Morning coffee/tea*

---

10:00 – 11:00 **B Brian Troutwine @bltroutwine - Getting Uphill on a Candle: Crushed Spines, Detached Retinas and One Small Step** 1. Alfa

*Speakers: Brian Troutwine*

Looking back through history, we often view NASA's early mission in terms of "getting to the Moon", discussing how this or that program served the purpose of answering Kennedy's challenge. This is wrong-headed. In this talk

I will discuss aeronautics research beginning with the Wright Brothers and ending with the first Shuttle launch in 1981. We'll see how NASA is an organization whose primary mission is basic research and development in aeronautics for the benefit of the public at large and space exploration. We'll see how the Lunar Program was a focusing of research to a practical, political aim which built off decades of basic research and necessarily side-lined other programs. It's my aim to convince you that Moonshot projects cannot be considered independently of their organizations and its history.

---

10:00 – 18:00

**Open Space**

6. Lobby

*Speakers: Open Space*

*Sponsors: 4Finance, Devbridge, Storebrand, Visma Lietuva, WIX Lietuva*

---

11:20 – 12:15

**B Osvaldas Grigas @ogrigas - Life without Objects**

5. Theta

*Speakers: Osvaldas Grigas*

Transitioning from OOP to functional style can be quite challenging, not least because OO programmers are used to thinking in nouns. Programming languages that use functions as primary means of abstraction force one to:

- design things in terms of verbs,
- find new ways of doing composition,
- rediscover polymorphism in a different light.

Fear not! What you have learned about good OO design can be applied to FP, in obvious and weird ways. Code examples will be presented in Clojure.

---

11:20 – 12:15

**I Amanda Laucher @pandamonical - Property Based Testing: Shrinking the Risk in Your Code**

3. Lambda

*Speakers: Amanda Laucher*

Perhaps you've been hearing a lot about Haskell programmers being absolutely certain that their code is correct but you haven't taken the leap into day to day Haskell development. Do not despair, there are techniques that can allow you to have confidence in your code without needing to change your development stack. In fact, you can use the same testing techniques Haskellers use without even using a language with a static type checker.

In this session we will be looking at Property Based Testing, and how this approach can allow us to avoid thousands of lines of testing code when ensuring that our code meets specification. Property Based Testing generates inputs based on the properties of the program that we stipulate, and so are based on the business logic, in the same way that types are. Furthermore, if the test fails, a good framework will shrink the problem to the smallest possible data set that gives an error, helping to pinpoint the bug. This session requires no previous knowledge of free-monads, co-products, or other terms you may have grown to hate.

---

11:20 – 12:15

**I Bozhidar Batsov @bbatsov - CIDER: Building a Clojure Interactive Development Environment that Rock in Emacs**

2. Beta

*Speakers: Bozhidar Batsov*

This talk is dedicated to CIDER - an interactive development environment for Clojure. CIDER is the most popular programming environment in the world of Clojure these days and it's quite unique, for it's not a plugin for IntelliJ, Visual Studio or Netbeans; quite the contrary - it's built on top of the Emacs editor and it still provides a ton of cool features that most IDE users have come to love and expect from a modern programming environment.

We'll discuss whether (why) it's a good idea to build dev tools on top of Emacs, the history of the project, its current state and the bold plans for its future.

Be advised - by the end of the talk you'll probably become an Emacs user for life, so attend it at your own risk! :-)

---

11:20 – 12:15

**I Chris Condrón @CLCondrón - Teaching my Team CQRS**

4. Zeta

*Speakers: Chris Condrón*

I first attended one of Greg's workshops on CQRS and message driven architectures several years ago and fell in love with the design patterns. However what seems so clear to me seems to often elude even senior developers who I try to introduce it to. The key element I've found in getting developers to write code using the new patterns is to get them thinking in the new patterns. Solving problems in the new way, rather than what they have always done. I'll be talking about the problems I've seen on different teams moving from a training context to solving problems in production code under a deadline. Then we'll cover some of the key mental blocks to adopting the new methodologies. Finally walking through successful approaches that have gotten people thinking and

solving problems in the new ways.

---

11:20 – 12:15	<b>I Venkat Subramaniam @venkat_s - Let's Get Lazy</b>	1. Alfa
	<i>Speakers: Venkat Subramaniam</i>	
	How can big data or highly responsive applications scale to the increasing demands for speed and short response time? Adding more servers to the cluster is not the answer. The smartness comes from being lazy as laziness can translate to efficiency and scalability. In this presentation we will learn about what lazy evaluation is, explore some data structures and APIs that promote lazy execution, and tie it back into scalability and efficiency.	
12:15 – 13:15	<b>Lunch</b>	1. Alfa
	<i>Speakers: Lunch</i>	
13:15 – 14:10	<b>B Mathias Brandewinder @brandewinder - Crunching through big data with MBrace, Azure and F#</b>	4. Zeta
	<i>Speakers: Mathias Brandewinder</i>	
	For data exploration and rapid prototyping, the productivity of an interactive scripting environment is hard to beat: simply grab data, run code, and iterate based on immediate feedback. However, that story starts to break down when the data you have to process is big, or the computations expensive. Your local machine becomes the bottleneck, and you are left with a slow and unresponsive environment. In this talk, we will introduce MBrace.net, an open-source and free engine for scalable cloud programming. Using the MBrace programming model, you can keep working in your beloved familiar scripting environment, and easily execute C# or F# code on a cluster of machines on Azure. We will focus primarily on live demos, from provisioning an Azure cluster with Brisk, to analyzing large datasets in a distributed fashion; in particular, we will discuss how this setup is relevant to data science and machine learning.	
13:15 – 14:10	<b>I Michael Feathers @mfeathers - The Slow Steady Industry Move Toward Tacit Programming</b>	1. Alfa
	<i>Speakers: Michael Feathers</i>	
	We're all aware that the industry is moving from Object-Oriented toward Functional Programming, but the move may be even deeper than that. As we adopt a strongly compositional style using tools like LINQ, Rx, Java Streams, and Ruby's Enumerable, we find that we approach a type of programming that is closer to what is common in the APL family of languages. This talk will explore the trend and its possible ramifications.	
13:15 – 14:10	<b>I Pavlo Baron @pavlobaron - Why monitoring sucks, and how to improve it</b>	3. Lambda
	<i>Speakers: Pavlo Baron</i>	
	Computers are good at solving recurrent problems. Much better than humans are. And still, we keep them dumb with a set of simplest heuristics when it's about monitoring complex infrastructures, leaving the largest part of the job - issue recognition and analysis - to ourselves. This might work with a server or two, but definitely won't in a larger setup, even if we convince ourselves it would. We need new approaches to monitoring our systems that combine the best of software engineering and mathematics. In this talk, I will explain the vision and the targets towards it.	
13:15 – 14:10	<b>I Pawel Sawicz @sawiczpawel - Mutate your code and reveal you true test coverage</b>	5. Theta
	<i>Speakers: Pawel Sawicz</i>	
	Session is about mutation testing, why and when you should mutate your code. What benefits comes by mutating your code. Simply test your own tests. It's very helpful tool with TDD where you are exposed to a lot of wrong assumptions and simple syntactic errors that can propagate other errors.	
13:15 – 14:10	<b>I Sam Elamin @samelamin - Monoliths to Microservices. A Journey</b>	2. Beta
	<i>Speakers: Sam Elamin</i>	
	Your monolithic system is a pain to work with and maintain. Moving to a distributed system will solve all your problems and you will be in developer heaven. Right? You will be working with cool technologies and amazing concepts. Plus, it's Microservices! So what could possibly go wrong?  In this talk Sam Elamin will relate his real life experience migrating a single ASP.NET application with a monolithic database to a distributed system dealing with £100,000 transactions every hour. Sam will cover the challenges faced, the lessons learned, and offer some final takeaways.  This "from the trenches" story will show you the pitfalls to avoid when doing Microservices.	

---

14:30 – 15:25	<b>B Kevlin Henney @KevlinHenney - Programming with GUTs</b>	1. Alfa
	<i>Speakers: Kevlin Henney</i>	
	These days testing is considered a sexy topic for programmers. Who'd have thought it? But what makes for good unit tests (GUTs)? There's more to effective unit testing than just knowing the assertion syntax of a framework.	
	Testing represents a form of communication and, as such, it offers multiple levels and forms of feedback, not just basic defect detection. Effective unit testing requires an understanding of what forms of feedback and communication are offered by tests, and what styles encourage or discourage such qualities.	
	What style of test partitioning is most common, and yet scales poorly and is ineffective at properly expressing the behaviour of a class or component? What styles, tricks and tips can be used to make tests more specification-like and can scale as the codebase grows?	
14:30 – 15:25	<b>B Yan Cui @theburningmonk - My adventure with Elm</b>	5. Theta
	<i>Speakers: Yan Cui</i>	
	Reactive Extensions (Rx) has brought reactive programming to the mainstream in recent years with successful adoption in languages such as C#, Java and JavaScript. But have you ever wondered what Rx will look like as a language?	
	Elm is a new programming language based on the idea of Functional Reactive Programming (FRP). Elm lets you create highly interactive web applications without all the messy callbacks tangling around shared states.	
	In this talk Yan Cui will give a gentle introduction to Elm and share his experience learning Elm and recreating Missile Command in Elm with less than 250 LOC. You will leave this session with a handle on the Functional Reactive Programming paradigm and a basic understanding of the Elm language.	
14:30 – 15:25	<b>I Ben Hall @ben_hall - Real World Experience Report on Running Docker</b>	3. Lambda
	<i>Speakers: Ben Hall</i>	
	Docker has taken the world by storm and is rapidly becoming the de-facto way to deploy applications and services. With a new development and deployment approach it brings new challenges and best practices.	
	During this talk Ben will discuss his experiences of working with Docker on a daily basis as a development platform and deploying it into production.	
	Ben discuss his experience with using Docker around areas such as:	
	<ul style="list-style-type: none"> <li>- Development, Test and Build lifecycle</li> <li>- Building and creating small, streamlined, containers</li> <li>- Auto-discovery architecture</li> <li>- Scaling production nodes</li> <li>- Resource management</li> <li>- Security concerns and considerations</li> </ul>	
	At the end attendees will understand the advantages along with the potential issues of running Docker based on real world experience. This should enable them to identify how to migrate and build their own applications using a container based architecture.	
14:30 – 15:25	<b>I Darach Ennis @darachennis - Our Little Pony</b>	4. Zeta
	<i>Speakers: Darach Ennis</i>	
	Pony is a relatively new LLVM based compiled language supporting ease of integration with native code. This talk takes a deep dive into Pony and the mature Erlang ecosystem. Plus, who doesn't like 'ponies'?	
	Talk objectives:	
	<ul style="list-style-type: none"> <li>- Learn a little about Pony itself, using Pony, and extending Pony with native extensions.</li> <li>- Compare and contrast to the Erlang ecosystem on a joyride through the fields of pony.</li> <li>- There will be pictures of ponies, natch.</li> </ul>	
14:30 – 15:25	<b>I Rachel Reese @rachelreese - Patterns and Practices for Real-World Event-driven</b>	

## Microservices

2. Beta

*Speakers: Rachel Reese*

At Jet.com, we've based our architecture around cloud-based event-driven microservices, and over the last several months, have schooled ourselves on what works and what doesn't. This session will walk you through the lessons we have learned on our way to developing our platform.

---

15:25 – 15:45

### Coffee/tea break

1. Alfa

*Speakers: Coffee/tea break*

---

15:45 – 16:40

### B [Dylan Beattie @dylanbeattie](#) - Are smart systems making us stupid?

2. Beta

*Speakers: Dylan Beattie*

"The Turing Test will be passed by 2020. Not by an advanced artificial intelligence, but by a human being who is stupider than their own phone"

Did you read about the man who drove his car into a lake because Google Maps told him to? Or the woman who put her phone into "airplane mode" and threw it out of a window? Does Google ever freak you out by showing you stuff it's not supposed to know about?

Software and smart devices are changing the world beyond recognition, and all too often, the human beings who create it are struggling to keep up. We create devices that can make crystal-clear hi-def video calls to anywhere in the world, and then laugh at someone who microwaves their iPhone because they read online that it would charge the battery. You spend \$800 on a tablet computer that doesn't even include an instruction manual - and then your three-year-old kid finds a shortcut for playing Peppa Pig videos that you didn't know existed. At the other end of the scale, we're building huge distributed systems too complicated for any human to understand. Decisions that affect our lives - the pages that show up in our search results; the people we meet on Tinder; the price we pay for car insurance - are being delegated to algorithms so sophisticated that nobody can explain why a particular result happened, or predict whether it will happen again.

So what can we do about it? As developers, how do we build systems that don't make people feel stupid? How do we empower users to make decisions and apply common sense in a world where tomorrow's technology is indistinguishable from yesterday's magic?

In this session, we'll talk about auto-correct, waterproof smart phones, cognitive bias, Markov chains, Windows 10, self-driving cars, chaos theory, the psychology of risk, Monty Hall, user experience design, the Dunning-Kruger effect, and why Facebook is still showing you adverts for cheap flights to Lithuania even though you're already here.

---

15:45 – 16:40

### B [Jeroen Soeters @JeroenSoeters](#) - The Hitchhiker's Guide To Neuroevolution in Erlang

3. Lambda

*Speakers: Jeroen Soeters*

Neuroevolution is a technique where we use algorithms inspired by nature to evolve neural networks. We will go on a journey on which we first explore the basics of a neural network, followed by looking at the beauty of evolutionary computation and ultimately go down the rabbit hole and combine the two to create a platform for evolving neural networks that can be used to tackle a wide variety of problems from cleaning robots to financial oracles.

---

15:45 – 16:40

### B [Pieter Hintjens @hintjens](#) - Ten Rules for API Design

1. Alfa

*Speakers: Pieter Hintjens*

Every software developer uses APIs and most of us make them. The design of a "good" API is a black art. You know one when you see one. And yet how many of us could explain why some APIs are complex and hard to learn, while others are clean, simple, and a joy to use. It's a question I'll answer in this talk, and provide ten rules for good API design.

---

15:45 – 16:40

### I [Jonathan Graham @graham\\_jp](#) - Reactive Systems: From Drug Development to Functional Programming

4. Zeta

*Speakers: Jonathan Graham*

Systems built as Reactive Systems are more flexible, loosely-coupled and scalable. This makes them easier to develop and amenable to change. They are significantly more tolerant of failure and when failure does occur they meet it with elegance rather than disaster.<sup>1</sup>

The approach to the design and development of manufacturing processes for the production of new drugs within the pharmaceutical industry has changed dramatically over the last decade. Focus is given to designing systems that are responsive to issues and constraints, through knowledge of the impact of exceeding standard operating ranges and the use of real-time analytics; resilient to failures that could occur at any point within the system; elastic to changing demands that occur during the lifecycle of manufacture through a flexible and well understood approach to scalability; and message driven, whereby the resources used and specifications required for a specific segment of the system are derived by the demands external to that segment. With Quality by Design<sup>2</sup> applied throughout the development process, the industry is now beginning to reap the benefits from the flexibility that Reactive Systems provide in production.

In this presentation we will use learning's from the Pharmaceutical Industry to explore the extent of the Reactive Manifesto for software development, and we will look specifically at how this relates to functional programming. The public demands high and consistent quality from the medicines that we take, and we should demand that same quality from the software that we develop. If you are passionate about the quality of your code, then this talk will provide you with a new perspective on how you think about your craft.

1 <http://www.reactivemanifesto.org>

2 [http://en.wikipedia.org/wiki/Quality\\_by\\_Design](http://en.wikipedia.org/wiki/Quality_by_Design)

---

15:45 – 16:40

**I Paul Stack @stack72 - Continuous Delivery - The Missing Parts**

5. Theta

*Speakers: Paul Stack*

A lot of developers have started to believe that hooking Visual Studio up to Azure and pushing code direct from their machines is CD. As much as I hate to say it, it isn't. Continuous delivery has so many more moving parts required to work together.

As we discuss concepts such as config management, orchestration, security, monitoring and logging, this talk will help developers realise that continuous delivery is something we need to continually measure, learn and adapt to make us a higher achieving organisation.

---

17:00 – 17:10

**Raffle time!**

1. Alfa

*Speakers: Raffle time!*

---

17:10 – 18:00

**B Mark Rendle @markrendle - Programming For The Criminally Insane**

1. Alfa

*Speakers: Mark Rendle*

Many programming languages strive to be expressive, succinct, elegant and performant.

Many others don't.

Guess which ones this talk is about.

---