Issue January 2017 | presented by jaxenter www.jaxenter.com #55

jaxmagazine

The digital magazine for enterprise developers

Featuring: jax FINANCE

Software ~~vs~~ and Finance

How digital transformation is reshaping the future of finance

Interview with Eric Horesnyi

"Banks understand that technology is no longer just a tool – but a core skill"

Payments are about to change

Providing alternative payment options – a thriving marketplace

Jump on the Java 8 bandwagon

Take advantage of the increasing use of Java in finance

Finance: What's in it for developers?

Does working in the finance sector pay off for software developers? The answer is "yes". IT is a must have for most banks these days – a massive change from its previous "optional" status. That being said, if banks are trying to win developers over, why shouldn't the latter learn a bit about this sector?

Finance and software engineering are not two parallel lines; finance needs software engineering's help to grow and keep pace with today's demands and competition. One popular opinion is that software engineers should become part of banks' core business and help transform the entire concept of banking. However, in order for that to happen, software engineers must not be treated as second-class citizens; development units have become a primary producer of business value, so why not admit that without them growth would be futile?

The technological needs of the finance industry include aspects such as Continuous Delivery, Java, Big Data, machine learning, low latency etc. By embracing these elements, the finance sector can change the status quo and co-exist with what appears to be a fierce opponent: FinTech. How should banks respond to FinTech? Acknowledge that disruption is here to stay, find ways to partner and transform challenges into opportunities. This way, the digital future can belong to both "traditional" banks and disruptors – provided that software engineering takes center stage.

If you want to hear more about the facets of finance and discuss openly about its challenges and opportunities, come to JAX Finance, which takes place in London between 3–6 April 2017.

Gabriela Motroc, Editor

Index

HOT

TensorFlow 1.0 brings machine learning to mobile devices

There's a new TensorFlow 1.0 release candidate and it comes bearing full-blown features. In addition to making it easier for Java and Python users to put TensorFlow to good use, the added support for iOS and Android has now been optimized. Speaking of which, this release candidate comes with new person detection and tracking demo implementing "Scalable Object Detection using Deep Neural Networks" (with additional YOLO object detector support), as well as new camera-based image stylization demo based on "A Learned Representation For Artistic Style".

JHipster 4.0 brings Angular 2 support

JHipster 4.0 is out! This release supports both AngularJS 1 and Angular 2.x and has successfully migrated to Yarn. JHipster is now the only available "full stack" generator for Angular 2.x and the only generator that creates non-trivial Angular 2.x code, in particular using their JDL Studio. Generation is now easier, faster, and safer as pain points have been removed from JHipster 4.0.

Kotlin 1.1 has reached beta

Kotlin 1.1 is not ready yet but it's coming "fairly soon". How do we know that? Kotlin 1.1 has finally reached beta. According to the official announcement, the biggest highlights of 1.1 are the following: coroutines on the JVM, JavaScript and Android, as well as full support of compilation to JavaScript.

An agile Java standard may not be possible

Simon Ritter, Deputy CTO at Azul Systems and alternate representative on the JCP EC wrote in a blog post after JCP executive committee's first face-to-face meeting that some changes will have to be made to the processes the JCP uses, otherwise the idea of making an agile Java standard is unlikely to materialize. Ritter explained that even though the fact that individual JEPs will be developed in isolation from other JEPs will enable developers to "take advantage of features as soon as they are ready", the following problem remains: how will this approach work in practice?

Oracle says goodbye to MVC 1.0

It's not necessarily bad news since Java Champion Ivar Grimstad is the new MVC spec lead. However, change is change: Santiago Pericas-Geertsen and Manfred Riem, former MVC spec leads, announced in mid-January that they have initiated a Transfer Ballot as per JCP 2.10 and proposed to transfer the MVC 1.0 JSR from Oracle to Ivar Grimstad. Werner Keil expressed his happiness for seeing "another JSR run by an individual".

NOT

©istockphoto.com/Nikada

This is why banks are reluctant to enter the roller coaster of FinTech

Banks are innovating but not at the rate you want them to

How can one enable innovation in banking? The answer lies in convincing the end client that innovation is safe and banks are not taking any risks. Of course this is easy to say but not that easy to execute.

by Todor Gigilev

Imagine you are in your late 80s, made millions in savings. Would you take bold steps and risk what you have already secured or will you just take time to enjoy your days? Now imagine you are part of the board of a multinational bank that has big market share and you are in charge of innovation. Would you push your people to come up with the most ground-breaking solutions or would you innovative just enough to keep your market share? This might give you an answer as to why many of the big names in banking still use mainframes and they are reluctant to enter the roller coaster of FinTech we have been witnessing the last five years.

Blockchain, NFC, Peer to peer lending are just a few of the options traditional banking could have fully adopted. This would have had tremendous impact on the way we exchange transactions, do business and live our lives. However, I cannot name a big bank that has jumped on the bandwagon and delivered a fully fledged product in this area. Just the opposite – the stories I hear are, for example, about two of the top executives of BNP Paribas in Bulgaria leaving the company to start their own Peer to peer lending platform called Klear.

Why didn't they initiate this project inside the organization? I think there are two factors causing this:

- Internal factor: Company culture in the traditional banking
- External factor: The public image of banks is all about security, while innovation relates to risk.

But before we delve into details, let's first define what is innovation in the eyes of a bank executive and what we, the "disruptive innovation" fans, picture in our minds.

What is innovation according to banks

I have heard stories that only changing the background color of the corporate banking portal took six months. That should be enough to set the tone of what you can expect when we talk about innovation in a large banking institution. So forget about authorizations of transactions through face recognition or personal branch in your home via augmented reality.

Last year, my company Dreamix developed a bespoke Java based software solution for a banking organization. It was considered innovative in terms of internal banking processes and how one can create new products (deposit or loans) without the need of any IT support.

Usually new banking products need a lot of IT involvement – for each new type of deposit/loan you need someone to implement tens of forms and wizards.

Instead, we managed to remove IT from the equation by implementing metadata driven framework for form generation. This way, operations can work directly with marketing and create different forms and wizards, without the need to call any IT personnel, which cuts the time for execution.

For some of you this might sound like something not that important, but if you need to wait six months for to change the background, imagine how much time you need to create a new product.

Company culture in banking

"Nobody has been fired because they use..." is a sales idiom used by large software vendors and it is a great epitome of the risk averseness in banking. The bureaucracy and the steep hierarchy in these organizations is transforming them into an uncompelling workplace for people who want to disrupt industries. Additionally, the attention is always on "not failing" rather than "implementing something new".

Innovation is always linked with uncertainty, banks are always linked with high security

Imagine you are about to become a client of the Imaginary-CryptoBank. How would you feel if you knew your deposit is turned into cryptocurrency and a distributed network is in charge of authorizing the transactions? You, being the reader of an innovation-oriented media, might be completely OK with this fact knowing that it will reduce the process time (real time transactions rather than twice per day RINGS transactions), the costs (thus bring down the taxes) and will be even more secured. However, chances are that your grandmother will be terrified if there would be nobody in charge of this network and everyone would be able to add a node and be part of this infrastructure. She will certainly prefer the old-fashioned bank that is not powered by the mechanism behind the dodgy "bitcoin" currency she is hearing scary news about on TV.

> *"Innovation is always linked with uncertainty, banks are always linked with high security."*

How to enable innovation in banking

I think the answer lies in convincing the end client that innovation is safe and banks are not taking any risks. Of course this is easy to say but not that easy to execute. We have a good example of consortium of banks coming up with a radical move to start a joint blockchain project. This way none of the big players risks their own reputation.

Another way to announce innovative projects is by strongly focusing on the physical design and digital UX of the innovation because, believe it or not, the mass client judges how reliable something is by the external look of it. For example, people facing a slowly loading website believe the end product is not stable. A good example here is Procredit Bank applying Digital Signage and educating their customers that technology is good for them.

Todor Gigilev has consulted organizations like Royal Bank of Scotland and BNP Paribas on their digital transformation software projects. He is one of the founders and currently CEO of Dreamix Ltd – bespoke software development company, that is delivering Java based enterprise solutions for reputable companies worldwide.

Interview with Eric Horesnyi

"Banks understand that technology is no longer just a tool – but a core skill."

Software is eating the world – according to Marc Andreessen. We are eating larger parts of the Finance cake – exclaim the FinTechs. There's a real revolution going on in finance and it's being driven by technology. FinTechs inspire the market while classic banks adopt more and more practices from the FinTech world in order to stay ahead of the curve. But what are the elements of the FinTech revolution? Eric Horesnyi, the whiz kid of startups, discusses with movers and shakers from the FinTech scene and gives you an insight into this revolution.

JAX Magazine: Payment, API-oriented business, DevOps and company culture, Crowdfunding, Big Data, and Blockchain are all key drivers of the FinTech revolution. So, let's go through these elements and talk about their impact on the finance industry. Starting with payment, where do you see the main movers and shakers – is it Apple, PayPal, who else?

Eric Horesnyi: Payment is the most mature sub-segment in FinTech, and the one that has been most covered and invested in over the past few years. There are thousands of payment-focused companies right now, including unicorns like Square or Stripe, and WorldRemit and Adyen in Europe. Maybe or certainly thanks to a regulatory environment that has become favorable, notably in Europe with the SEPA scheme that has created a vast environment for open competition with the European Passport, fostering innovation, to the benefits of citizens. Payment market has gone from domestic to European, without having to adapt core systems to each country.

Alipay, Apple Pay and Google Wallet now validate the market rather than crush it, and demonstrate that the Web Giants can enter banking markets, leveraging not only their technology stack, but also hundreds of millions of banking information they already have in stock with their core activities.

JAXmag: To which degree do you think the traditional banking world is prepared for a tectonic shift in retail banking?

Horesnyi: Traditional banks have now understood that their competitors in the long run could be the following: web or telecom giants with a profitable non-banking core business or niche fast-growing ventures disrupting – or using sharding, one of the web practices – their business models.

They have all understood that technology was no longer just a tool, but a core skill. However, only a few have come to consider technology an enabler as important as their branches used to be. Not many have started to consider UX as their main objective to retain and grow their business. Many remain entangled with legacy systems that seem expensive, inert – as in full of inertia, versus agile –, and closed. Migration to new UX standards set by Amazon or Netflix requires investment and courage, but most importantly a realization that the technological paradigm shift – Continuous Delivery, Microservices, DevOps – has empowered a new generation of user experience that is not just a gig, but a complete culture.

Portrait

Eric Horesnyi is CEO at Streamdata.io. He is a High Frequency Trading infrastructure expert, passionate about FinTech, IoT and Cleantech. Eric has worked in San Francisco, New York City, Mexico and now Paris.

Traditional banks feel threatened in their core business models by digital giants that have access to hundreds of millions of user information they already monetize, including banking information. It must feel like the scene of invasion in the Pixels movie, best illustration I know of "Software is eating the World".

JAXmag: Moving on to the API-oriented business – do you anticipate that traditional banks will be deconstructed and that small, independently operating "Microservice-Banks" will unite to provide a consistent service?

Horesnyi: It is very hard to follow and define a strategy against thousands of players coming from behind. Traditional banks might compare this scenario with Gulliver traveling to Lilliput.

Banking is one of the last industries (governments and administrations too) to be impacted by the Web. Some mention the Third Industrial Revolution, I am not qualified to confirm, but for sure, after the revolution hit the music and video industry with streaming, retail with e-commerce, education with MOOC etc. there is no reason for banking not to be turned upside down and reborn into models hard to imagine ten years ago.

The finance business has always been fragmented, full of coopetition. With companies selling services to each other while competing on another market, the revolution that FinTech brings has inserted itself quite naturally within the "traditional" landscape. And APIs have turned to be the best way for FinTech to collaborate amongst themselves, to leverage latest contributions in Analytics and UX, and even to provide services to traditional banks.

APIs were invented to facilitate programmatic open collaboration and reuse between software teams. Google and Amazon APIs have in fairly recent years demonstrated that APIs can be a core business. With more than 16,000 APIs now in ProgrammableWeb, and just 2,000 more in 2015, it sounds like APIs are more than a fashionable trend. APIs have become the new store front.

JAXmag: Let's remain for a second at Gulliver being a symbol for the big banks, chained by the small and agile inhabitants of Lilliput. What can large organizations do to be prepared to become as agile, flexible, and customer-oriented as the smaller challengers? And what role does technology play in this situation?

Horesnyi: Technology remains a tool. Best use of technology actually requires a certain culture. Lilliputiens today are API and UX providers focused on a niche market opportunity. The obvious way to obtain this agility are Microservices: 2-pizza rule teams (6 people) with a responsibility on a feature, self-contained infrastructure – independent from Cloud location – exposed through APIs, streaming their states to other microservices so that the entire system can stay tuned (as we did with ESB and Middlewares before within data centre).

To get from a Monolith architecture to a microservice one is a journey. And that path is not necessarily the same for all organizations.

Once Microservices are in place, you can imagine that a team focused on a KYC feature may decide to either scale down and sell their APIs to other companies, or shut itself down and invite others to use a third-party API doing a better job at KYC.

> *"Payment is the most mature sub-segment in FinTech."*

Another implication is that the famous Legacy issues can be circumvented and prepared to be replaced without too much impact on the entire system, other than a positive impact to open it. Legacy systems are treated like Sacred Monuments, to which a common interface is imposed (Rest API ideally) to make it work with other components.

JAXmag: You have mentioned the tech element in "FinTech": To what degree does such a company culture interact with technologies in use, e.g. Clouds, Containers, Microservices?

Horesnyi: Actually, the financial sector is the most technology-intensive industry already. It comes from the fact that finance is a proposition that is immaterial.

Finance (from fidere, trust in latin) is all about trusting somebody else to accomplish what you expect for the future:

"APIs have turned to be the best way for FinTech to collaborate amongst themselves , to leverage latest contributions in UX and analytics, and even to provide services to traditional banks."

keep my values, transmit some values, or grow my assets. Manipulating such virtual concepts (against building a car or growing crop) has made finance a heavy IT user since computers were born.

FinTech is about bringing the dev culture into the financial sector. Technology today is not a series of protocols and norms the Web has helped build for twenty years. It comes with a culture of balanced collaboration, contributions, efficiency, openness, Darwinism, pragmatism and more (I am not a sociologist, but I believe our dev audience will recognize itself). Using technical words, that is the Microservices (2-pizza rule definitely addresses Dunbar's number!), continuous delivery with containers (to ship features – hence the term, Docker motto is build>ship>deploy – asap and get feedback from users with A/B testing), Agile/Scrum (roadmap decisions are too important to be decided top down!), DevOps (the link between the virtual world of code and real machines to action it all), Cloud (it is possible to safe harbor in the cloud), APIs (the new front stores from a dev team to another), event processing and streaming (critical for real-time programmatic reactions), Analytics and Machine Learning (Artificial Intelligence-Neuron Networks and Graph Theory generating money outside of the algorithmic trading sector).

JAXmag: Then comes crowdfunding – which seems to be the less technical factor in our previously defined list of key elements of the FinTech revolution? Does Crowdfunding compete with traditional banking, or is it just complementary?
Horesnyi: Crowdfunding is quite visible compared to its size. It includes Crowd Lending (someone/a company wants to borrow, take a loan) and Crowd Equity (a company raises capital for a project, against a percent share of the company). Lending Club – a unicorn – is Crowd Lending, and yes, banks in the USA have been shaken by this risk-minimized/scored peer-to-peer model to the point where some banks invest in Lending Club sizable amounts now. Crowd Equity like Kickstarter and Angel List have had more coverage, as they are as much marketing tools (to recruit first users) as financing tool (especially for seed, but exceptions of multi-million funding like Pebble are growing).

JAXenter: Do you think banks will fight back or they will somehow jump on the bandwagon like in the case of Lending Club?
Horesnyi: Banks will jump on the bandwagon, either by contributing lending or capital to the platforms, or taking shares into these ventures.

JAXmag: Data, as the raw material of so many businesses shows, is also going to play a crucial role. Let's talk about automated guidance for customers, the so-called robo-advisors. Can you explain this phenomena?
Horesnyi: Robo-advisors represent the use of algorithmics to support financial decisions to grow your objectives. Human fund managers are being challenged by algorithms: three-fours of trades are already algorithmic in Forex, best funds – e.g. Renaissance – have been algorithmic for the past twenty years, and people have followed ETF strategies since Barra & BGI in San Francisco created the concepts twenty years ago (idea: you take less risk by investing in baskets of shares than individual shares, you choose your strategy and pay less in management fees). I am a big fan since Mint, now Wealthfront, Betterment, and Personal Capital. Give your time horizons, your household projects, get your risk-aversion assessed and boom that's it: "Your financial health is built more smartly than median financial advisor and for less."

JAXmag: Understanding your customer better, their needs and preferences also falls into the UX category. This game is mainly controlled by Google, Facebook, Apple & Co. Can finance still catch up or will the future customer only be known by the large internet companies mentioned above?
Horesnyi: Any company can always catch up on UX: just listen to your customers and consider it more important than anything -than politics, or your personal exposure in the organization. It is not impossible. Without having worked there, so this is not a recommendation, I would say that a bank like ING in Europe has started to right-track on the IT front five years ago, and Capital One amazes me by the quality of people they keep recruiting from Google, Facebook, Amazon, Apple & Co.

Interview with David M. Brear

"Technology not going with finance is like saying macaroni doesn't go with cheese."

David M. Brear, renowned speaker and CEO and co-founder of 11:FS, believes that banks can and should put the current technological innovations to good use. We talked to Mr Brear about the FinTech movement's "disruptive" label and whether banks will "evolve or die." In this interview, Brear, weighs in on the inevitable clash between FinTech and the banking industry, banks' next step and what happens if they miss the boat and not adopt new technologies.

JAX Magazine: How can one innovate in the financial space?
David M. Brear: The same way you innovate in any existing market, you listen to customers, you look for unmet needs or unsolved problems, and you create and execute a great solution. The digital transformation happening now has not really affected banking yet. At 11FS, we believe digital banking is only one percent done. The products of today are simple, blunt and from another era.

The financial space brings extra regulatory demands due to the fact that you are likely handling money or information about money. This makes the barrier to entry higher, but thankfully in some markets we are seeing regulators drive innovation. The FSAs Project Innovate in the UK is an excellent example.

JAXmag: What's the difference between failure in fintech and failure in finance?
Brear: I think there are parallels as all FinTech companies need to make money to exist. A failure in their technology will eventually lead to a failure in their finance. This applies the other way round also.

Technology can make things quicker, cheaper, and easier. Used badly, like any tool, it can also make things slower, more expensive, and complex.

If you are billing yourself as FinTech, then you need to show your technology credentials. There have been some notable security lapses in the FinTech world of late. How a company recovers from that both technically and culturally will tell you a lot about any company.

> *"Technology can make things quicker, cheaper and easier."*

JAXmag: Still, some people claim that technology does not go well with finance. Should we blame technology for these unfortunate events?
Brear: Frankly, no! Technology not going with finance is like saying macaroni doesn't go with cheese. From earliest forms of physical money where people built in mechanisms for spotting fakes to today – where we try to use blockchain to rethink the sins of technology forefathers – Tech and Fin are completely intertwined. I'm a big believer (in the same way they say there is no such thing as bad children – just bad parents); I do not think the banks have a technology problem, they are just terrible at times at putting it into action. Technological innovations will be the heart and blood of the banking industry for many years to come and if big banks do not make the most of it, the new players from FinTech and large technology companies surely will.

JAXmag: Should banks be the ones to take the lead and ease the transformation towards the future and BaaP (Banking as a Platform)?
Brear: I'd see this as a great way to maximize the opportunities that the banks have and the new players don't in a sig-

Portrait

David M. Brear is CEO and Co-Founder of 11:FS, a consultancy that helps banks be truly digital. He's also co-host of the top-rated business podcast FinTech Insider.

"Technological innovations will be the heart and blood of the banking industry for many years to come and if big banks don't make the most of it, the new players from FinTech will."

BaaP play interesting from a partner's perspective but also a sustainable (if they do it right!) advantage. I think the hardest part in this for them to get over is going to be the cultural barriers that such a shift in sharing and working will bring to make BaaP work. If I was a bank CEO right now I would be setting up a team of people to make this happen along with the traditional setup.

JAXmag: Do you believe the feud between traditional banks and FinTechs is real or is it just a stunt performed by one of them?
Brear: Yes, it's real because lots of people are using FinTech players. The banks last year have tried to move the dialogue to one of collaboration rather than competition as I think anyone would try to switch the narrative in a losing PR battle. The big shift was that – I think – the current biggest battle is actually between more the traditional suppliers and FinTech than the Banks and FinTech. Companies entering markets and being able to provide services that the big companies cannot offer, or worse, the ones they can for significantly cheaper, has really had an impact on the industry and I think will be a trend that will continue on.

JAXmag: Is FinTech a global pandemic?
Brear: Without doubt it is, yes! It's the cool cousin of banking that everyone wants to be involved in. I think we are at a real tipping point with it. We are seeing the excuses for banks providing poor customer experience losing credibility and in a market where technology players are providing personalized and contextualized experiences into the daily lives of all walks of life then how long can banks continue to underwhelm?

JAXmag: Do you think the phrase "evolve or die" describes banks' situation right now?
Brear: Sure. Evolution for me though is clearly optional. Not all banks are going to survive in the way that they have in the past. The thing about evolution is that it works one death at a time in the wild while in banking its really one retirement at a time. The cultural shifts required and the adoption of new and challenging technology is beyond where most comfort zones are.

JAXmag: Does FinTech still deserve the label "disruptive" or has it outshined this tag?
Brear: I think the term disruptive has been overused a lot in the last five years. If something isn't being disrupted, maybe you shouldn't be doing it? Bullshit! People like to attach silly titles to things!

Keynote

DevOps Kaizen: Empowering Teams to find and fix their own Problems

Damon Edwards (SimplifyOps)

We all love the aspirational DevOps talks about companies achieving blistering speed and dazzling nimbleness. But improving your own organization's performance – from where they are now to performance levels equal to the industry leaders – seems like a very long and difficult road. What is missing in most organizations? A repeatable system that empowers teams to find and fix their own problems. This is a prescriptive talk about empowering and transforming organizations using a methodical — and totally reasonable Kaizen (Continuous Improvement) approach. We'll look at ways of combining known techniques like value stream mapping, Lean waste analysis, and improvement kata in order to fix organizational, process, and tooling problems. This talk isn't about mythical silver bullets or vague philosophies. This talk is about taking a fresh look at proven Lean techniques that already work in high-performing IT organizations.

Damon Edwards (SimplifyOps)

Damon Edwards is a Co-Founder and Chief Product Officer of SimplifyOps, the makers of the popular job scheduling and runbook automation tool, Rundeck. Damon Edwards was a co-founder of DTO Solutions, a DevOps and IT Operations improvement consultancy. Damon has spent over 15 years working with both the technology and business ends of IT operations and is noted for being a leader in porting cutting-edge DevOps techniques to large enterprise organizations. Damon is also a frequent conference speaker and writer who focuses on DevOps and operations improvement topics. Damon is active in the international DevOps community, including being a co-host of the DevOps Cafe podcast, an early core organizer of the DevOps Days conference series, and a content chair for Gene Kim's DevOps Enterprise Summit.

CONFERENCE TRACKS

- Software Architecture
- Core Java & Programming
- Continuous Delivery & DevOps
- High Speed & Low Latency
- FinTech & Innovation
- Fix & Messaging
- Big & Fast Data

Workshops

Applying DDD at pace for Effective Microservices

Russ Miles (Atomist)

In this workshop Russ Miles will demonstrate and implement the DDD patterns that are applicable to building great micro-service-based systems. Pulling from years of experience building this style of systems, this workshop will share the deep technical design skills that help make your microservices adoption a success.

Topics Covered

- Design a monolith ready for microservices
- Apply patterns such as Bounded Contexts, Events and Event Sourcing to be flexible enough to embrace microservices at the right point for your application
- How to build fast data flows and integrate with big data tasks from your microservices.
- Deploy, Upgrade, Stress and Manage your microservices so that they are ready for production

Docker in production

Gianluca Arbezzano (CurrencyFair)

Docker 1.12 offers a built-in orchestration manager. This is a big step because allow us to use Docker to manage our production environment and not just the runtime engine for our build containers. This workshop is designed for sysadmins, DevOps and all engineers that already know what a container is and the basics about the Docker ecosystem.
After a deep drive into the main concepts of Docker Swarm, we will create an infrastructure environment made of 5 nodes and we will install few tools to monitor, scale and make secure our environment. We will use all best practices to make our environment secure, stable and also to understand how to manage a production environment. Continuous Integration, Docker Registry, Monitoring, Logs and so on. Attendees will leave understanding what Docker offer to build a solid production environment for our applications and microservices.

Web Hacking: Pentesting and attacking web apps

Christian Schneider (Freelancer & White-Hat-Hacker)

In this hands-on workshop we'll all attack the training web app, to take on the role of a pentester one step at a time. You'll learn how to work with professional security tools through a range of practical tasks and will also learn pentesters' general approach for attacking web apps.
Of course, we'll also deal with defensive measures for protecting the security holes found, though our focus will remain the systematic use of professional hacking tools for carrying out (partially automated) security analyses. Once you've completed this workshop, you'll have practical experience of carrying out attacks on web apps, which you can transfer into your own software development work so as to increase the security of your projects for the long-term.

Russ Miles (Atomist)

Russ' experience covers almost every facet of software delivery having worked across many different domains including Financial Services, Publishing, Defence, Insurance and Search. Russ helps to change all facets of the software delivery process in order to remove unnecessary and costly complexity in everything from developer skills and practices, through applying the right processes for the job at hand, to ensuring that the right change is delivered, be it through software or otherwise. Russ Miles is an international speaker on techniques for achieving the delivery of valuable software as well as a published author, most recently of "Head First Software Development" from O'Reilly Media. Also author of "Antifragile Software", which is available on LeanPub.com.

Gianluca Arbezzano (CurrencyFair)

Gianluca Arbezzano is Software Engineer at CurrencyFair, a tech financial company. He works on different layer like automation, scalability and distributed system. He also is Open Source contributor and Docker Captain, strong believer in best developing practices and supporter of different User Groups.
His fields of interest are various and constantly in evolution: in the last year, he worked a lot on scalable infrastructures, reaching the goal of building some of them on top of AWS, DigitalOcean and OpenStack.

Christian Schneider (Freelancer & White-Hat-Hacker)

I write software since the nineties, work as a freelance software developer since 1997 (with Java since 1999) and focus on IT-Security since 2005.
Aside from the traditional software engineering tasks I support clients in the field of IT-Security. This includes penetration testing, security audits, architectural reviews, and web application hardening. Several times a year I conduct inhouse training courses on topics like web application security (focussing on Java) as well as on SecDevOps concepts for bringing security into agile projects.
Sometimes I enjoy writing articles about web application security and speak/train at conferences about web application hacking and hardening. As an Advisory Board member of JAX 2014, WJAX 2014 and JAX 2015 developer conferences responsible for their Security Days I constantly try to guide developers to include security aspects in their projects.

Sessions

Turbo Charge CPU Utilization in Fork/Join Using the ManagedBlocker

Dr Heinz Kabutz (JavaSpecialists.eu)

Fork/Join is a framework for parallelizing calculations using recursive decomposition, also called divide and conquer. These algorithms occasionally end up duplicating work, especially at the beginning of the run. We can reduce wasted CPU cycles by implementing a reserved caching scheme. Before a task starts its calculation, it tries to reserve an entry in the shared map. If it is successful, it immediately begins. If not, it blocks until the other thread has finished its calculation. Unfortunately this might result in a significant number of blocked threads, decreasing CPU utilization. In this talk we will demonstrate this issue and offer a solution in the form of the ManagedBlocker. Combined with the Fork/Join, it can keep parallelism at the desired level.

Data Hunters: The Rise of Quant Consultants

Pierce Crosby (StockTwits)

Demand-based data consultants have taken the helm of the emerging data ship. The rise of alternative data sources and the slowness of incumbent institutions has given room for these specialists to source, develop and redistribute new forms of signal. In this talk we give an overview of data consultants, the gap between quantitative and quantimental, and the value proposition of going 'full stack'.

PSD2 and Open Banking UK: Providing Banking APIs in Times of Change

Stefan Weiß (Fidor Solutions AG)

Fidor Bank was the first real bank to offer open RESTful APIs with OAuth2.0, in order to give bank customers and partners access to banking services and customer base. New regulation like EU's PSD2 and UK's Open Banking now forces banks in many countries to provide public APIs compliant to regional flavors – most of them are still moving targets. Stefan Weiß, responsible for the API product management at Fidor Solutions will talk about the challenge to provide a global platform that is not only compliant to current and upcoming regulation but also provides opportunities to build new business based on APIs.

Social Media, Real Time AI and the search for Alpha

Dr. Jamie Allsop (TP ICAP)

The continual search for alpha in trading today has lead to an explosion of interest in alternative information sources, such as social media (Twitter, Stocktwits and so on), and the use of artificial intelligence to make sense of it all. In the world of conversations and opinions it's hard to sift through the noise and filter out the real events that are impacting the markets. Data mining, natural language processing and heavy retrospective analysis has shown there is valuable information, but there are significant challenges finding that information in real time. Working with Yedup we've built an AI system that can extract real-time alpha and provide an API feed of hundreds of stocks and their related companies and industries. Used by one of the worlds biggest market makers we talk through some of the architectural choices and approaches needed to realise the real-time AI found in the Yedup system.

Strata, the Open Source Java Library for Market Risk

Stephen Colebourne (OpenGamma)

Strata is the new Open Source library for market risk from OpenGamma, written in Java SE 8. It follows the many of the design principles of Joda-Time and JSR-310, bringing easy-to-use and high quality open source to the domain of quantitative finance and market risk. In this talk I will show how the library allows you to price financial instruments like interest rate swaps and options, demonstrating the key elements of the library including curves, conventions and holiday calendars. Join me to explore the future of open source market risk.

From HFT to Laplace Demon, when timed data technology curves the market

Eric Horesnyi (streamdata.io)

The race for low latency data continues. 10 years ago, Flashboys were helping HFT make money with low-latency infrastructures. Today, hedge funds build AI brains pumping hundreds of sources of data in real-time, seeking ubiquity to build Laplace Demons.

	EARLY BIRD (until Feb. 23 incl. Team Discount)	EARLY BIRD (until Feb. 23)	STANDARD PRICES
Main Conference (2 Days) + Power Workshops (2 Days) I 3 – 6 April 2017	£ 910	£ 1299	£ 1399
Main Conference (2 Days) + Power Workshop (1 Day) I 3 – 6 April 2017	£ 770	£ 1099	£ 1199
Main Conference (2 Days) 4 – 5 April 2017	£ 525	£ 749	£ 799
Power Workshop (1 Day) 3 or 6 April 2017	£ 315	£ 449	£ 499

All Prices plus VAT

Providing alternative payment options – a thriving marketplace

Payments are about to change

No matter how you change the wiring, there is one major issue – ultimately, you need to take money from one place, a bank, and pay it into another, another bank. Still, the world is changing and alternative payment options are springing up like mushrooms after the rain. Let's see where they are heading.

by John Davies

Payments, as the name would suggest, means taking money (physically or electronically) from one place and paying it to another. We usually think of it as card payments, physical or on-line (eCommerce) but it can also refer to mobile wallets, direct bank-to-bank transfers for salaries and even crypto-currencies. However you think about it, payments represents one of the largest marketplaces on the planet.

Visa, for example, handles over $5 trillion in total global volume annually. At two percent per transaction, that's $100 billion a year, but most of that two percent goes to the issuer (the bank name on your card). Visa takes just a few basis points (bps) [1] to provide the "plumbing" that connects all the banks together. In fact, there are several roles in a typical payment, the cardholder (you and me), the merchant (who you're buying from), the acquirer (the company providing the POS [2] system), the network (Visa in this example) and the issuing bank (the bank that provide the cardholder with his/her card). For the merchant to get money from the cardholder, the acquirer, network and issuer act as intermediaries, taking little bits of that two percent off the transaction. It can get more complex as we add affiliates, foreign exchange (FX) and sales commission and more. As you can imagine, with tens of billions of dollars flowing through networks like Visa every day, just 1bp is worth millions. Hence, there is a thriving marketplace in providing alternative payment options.

Legacy and regulation

No matter how you change the wiring, there is one major issue. Ultimately you need to take money from one place, a bank, and pay it into another, another bank. Banks are legacy institutions and their technology reflects that. While almost every investment bank on the planet uses Java on Linux, they have literally hundreds of systems, many from previous mergers or integration into systems that are geriatric. At one major bank where I worked, we found over 40,000 database instances. Even the smallest project costs millions in such an environment, not just for the PowerPoint jockeys, but the developers who must integrate into over a dozen databases and systems. Banks find it extremely difficult to adapt to new payment methods or provide simple and cheap APIs.

Additionally, at least since 2008, compliance and regulation has buried most banks to the degree that they cannot escape technical debt, and find themselves slowly imploding. The latest directive is PSD2 (Payment Service Directive), looming as a significant headache for the banks to implement.

A payment – step by step

We can split payments into two parts: 1) acquiring, i. e. taking consumer funds for merchants, and 2) disbursements, ie. paying funds from merchants out to their supply chain, salesforce, etc. While closely linked, they usually take place in two distinct parts. When buying something on Amazon, money is credited from a consumer account and settled to Amazon. Then, Amazon pays the vendor separately. There is a good reason for this, and it's all about compliance and regulation.

If you were to pay the vendor directly your bank would need to know where the money's going. Equally, if the vendor received large amounts of money from seemingly random accounts, their bank would also need to know where it's coming from. With Amazon acting in the middle, we simplify some of the compliance and Anti Money Laundering (AML). Amazon can also manage the FX by presenting the price in the buyer's currency and choosing how to pay.

When a consumer pays Amazon, Amazon's merchant acquirer, like Chase Paymentech, asks your issuing bank to credit the money from your account, and settles it to Amazon. Amazon then instructs the vendor to ship the goods. Amazon actually takes ownership of the goods for a split second to meet regulatory rules. Amazon, at a later date, then tallies up the goods sold by the vendor, subtracts commissions etc. and instructs its bank to pay the vendor in his/her currency.

When the vendor doesn't receive the correct amount of funds, we need to involve the bank. Amazon's bank supports Amazon's needs because of billions of dollars deposited in accounts. When Amazon contacts its bank, the bank has to trace the payment across dozens of databases, logs and messaging systems. Worse still, it usually then goes on to another bank, a corresponding bank, and/or the vendor's bank. Typically about one

percent of payments fail, most errors are related to on-boarding or the vendor putting in the wrong details, not Amazon or the bank. This is made worse in the US where the banking system still relies on cheques. It costs the banks millions to service these payments, but they do it to keep the lucrative account.

I'll drop the Amazon example from here, the same is true for Airbnb, Uber, Lyft, Apple, governments, employers, anyone making mass B2B or B2C payments.

Accounts aren't exclusive banks

We all know that the emerging market in Asia is a gold mine, the problem is that many of them don't have bank accounts. Shock, horror you might say, but in this day and age you can survive very well without an account. We have hundreds of millions on AliPay in China, M-Pesa in Kenya and several other countries in the area, heard of PayPal? My own children have a card called "Osper", they have current and savings accounts. My wife and I can control where they can spend their pocket money and they don't have to carry cash when they shop on their own for expensive items like shoes, it's all managed from an app in realtime. The problem is, none of these are bank accounts that a business can pay from their bank accounts, notwithstanding the fact that these modern types of accounts are only going to become a larger and larger part of the global economy.

Crypto-currencies are not the solution

Whether you're an anarchist or banker, you will have surely seen the hype around crypto-currencies, with Bitcoin at the forefront. I own some Bitcoins, I've had some since the very earliest days and have made quite a bit of money from them. But that was purely luck. I would have made the same from buying gold or Apple shares at the right time. Bitcoins are a minute part of the market, used predominately for porn, drugs and laundering. You can use them to buy coffee and cakes in some places, and you can even use Bitcoin ATMs in modern cities, but it's just a research experiment that a bunch of anarchists took seriously.

Firstly, it takes a good ten minutes to validate a transaction, by which time your coffee will be cold. It's very limited in what can be transacted due to the 1k limit and the only people who can change that are the few (I believe 2) developers who have commit writes to the code. Let's just say it's in the interest of many of the competitive crypto-currencies to make sure it doesn't change. Bitcoin is not controlled by a rich, over-paid banker, backed up by a central bank, a democratically elected government and an army, but rather a few programmers. Where would you put your money?

Blockchain – today's technology hammer

Blockchain is not the solution to everything, the idea is not new, ledger "checksums" have been published in newspapers for decades, trying to destroy every single copy of the New York Times a week after it's been distributed across the globe is about as challenging as reverse engineering the hash on a blockchain. A ledger is a ledger, yes it could well be on a blockchain or it could be in a database, databases can still be encrypted and can also have checksums. As an architect, I like to solve problems with the right technology not use a technology to find problems. There are a few interesting features of a blockchain such as smart contracts but all of the traditional problems of CAP theorem etc. still apply to blockchains, enthusiasts seem to forget this. One thing is true though, the only blockchain worth using is the permission centralised or de-centralised model, as the distributed and un-permissioned model (as used by Bitcoin), creates more problems than it solves.

So what is the solution?

Payments will not change over night, you can't simply re-route trillions across a new system and change the entire eco-system. We're talking about so much money that it could totally change the global economy in a major way. Traditional banks struggle to survive, we've seen a lot disappear over the last few years and more will follow. I predict large changes when the US tax model changes and companies are able to repatriate their money back to the US, what will happen to the banks holding those trillions of US dollars today, how will they remain viable under current regulation? Brexit will have a major impact on many of the UK banks previously able to count European holdings in the same pool. We're in for a rough ride in the next few years. Opportunity for some but for many it will be a tough dose of reality.

We will see some of these new accounts and banks starting to provide better and better services. More and more people will start to use apps on their phone to receive salary, make payments and purchases. Some of the more agile banks (almost a oxymoron) will realise this and strip off their retail arm, they may partner or fund one of these new accounts. We see the start of this already with banks closing down high street banks. They will then be far more competitive in providing services such as savings, investments, loans and mortgages.

What am I up to?

Having just sold my last company, C24, that provided much of the integration for over 30 of the world's largest banks I'm working on what I believe will be the catalyst for much of the change I predict above. I'm obviously not alone, I'm working with some of the sharpest people on the planet, several of the large banks, one of the new banks I mention and some uber-personalities in this space. You will hear more, it's not a secret, now is just not the right time to write about it. Of course I will be speaking about this over the coming months.

John Davies is co-founder and CTO of C24, a London-based fast data company specializing in high-volume, low-latency complex messaging. With customers including many of the world's largest investment banks, C24 provides data optimisation for standards like SWIFT, ISO-20022, FpML and FIX as well as proprietary formats. C24 has recently released a new data optimization product – PREON – that creates highly optimized binary versions of these complex messages reducing memory and network usage by over 20 times, while significantly increasing performance. John has been global chief architect at JP Morgan, BNP Paribas and was the original architect behind Visa's V.me (now Visa Checkout). John has co-authored several Java books and is a frequent speaker at technical and banking conferences around the world. He is married to a French wife and has three boys (12, 14 and 17) who all love traveling (as long as there's internet).

References

[1] A basis point (bp) is 1% of 1% so 20bp is 0.2%. bp is often pronounced "bip"

[2] POS: Point of Sale

Interview with Paolo Tasca

The traditional banking sector is much more "intrusive" than Bitcoin

The Bitcoin economy has gone from an early prototype stage to a second one characterized by gambling and black markets and finally to a third stage which steers away from "sin" and toward legitimate enterprises. This is the conclusion of a research paper titled "The Evolution of the Bitcoin Economy: Extracting and Analyzing the Network of Payment Relationships."

Economists Paolo Tasca, Shaowen Liu and Adam S. Hayes have joined forces and analyzed the evolution of the Bitcoin economy. The research paper published last year unearthed some interesting facts about where this cryptocurrency is going and what role the blockchain technology will play in our highly digitalized future. We talked to Paolo Tasca, senior author of "The Evolution of the Bitcoin Economy: Extracting and Analyzing the Network of Payment Relationships", and director at the Centre for Blockchain Technologies at University College London, about the paper and the future of both Bitcoin and blockchain.

Portrait

Paolo Tasca is a FinTech economist specialising in P2P Financial Systems and Systemic Risk. He is the co-author of the "FINTECH Book" and the co-editor of the book "Banking Beyond Banks and Money". A former senior research economist at Deutsche Bundesbank, Paolo is now consulting different international organizations including the EU Parliament on blockchain technologies. Paolo is also a Research Fellow at CFS, Goethe University and a Research Associate at the Systemic Risk Centre of the LSE. He holds an M.A in Politics and Economics (summa cum laude) from the University of Padua and a M.Sc. in Economics and Finance from Ca' Foscari, Venice. He did his PhD studies in Business between Ca' Foscari Venice and ETH, Zürich.

JAX Magazine: How did you decide to start this paper? What is its goal?

Paolo Tasca: Differently from other datasets, all the transactions in public blockchains like Bitcoin are publicly available. This means that one can easily observe them under microscope. On the other side, all the originators and receivers of the transactions are hidden behind pseudonymity.

Thus, for us it was interesting to develop a new methodology (Pure User Group analysis and Transaction Pattern analysis) in order to extract the network of interdependence among the biggest economic entities which, although being largely anonymous, we categorize by their business line. Our method can be exported to other blockchains (like Ethereum) and can be used to run stress tests analysis on the given networks.

> *"Bitcoin is as safe as you can get thanks to the Blockchain and encryption."*

JAXmag: In the paper you mentioned that you can map some of the activity and interaction among Bitcoin users because pseudonymity and not strict anonymity is what characterizes these exchanges. Does the idea of pseudonymity ruin one of this cryptocurrency's primordial goals – namely to be anonymous?

Tasca: It all depends on what your goals are in being private with your money. The traditional banking sector, along with credit cards, money transfer and electronic payments such as PayPal are all much more "intrusive" than Bitcoin. Your credit card company knows more about you than Google might. Cash is also not by definition wholly anonymous since

"Our socio-economic systems will probably adopt different Blockchain infrastructures which will coexist together. New opportunities will emerge for other sectors to redesign their business logic."

if I buy something from you in a shop for $5 our identities are revealed to each other. That said, there are methods (e.g. Coin Join) to make bitcoin transactions more private, and some digital currencies, such as Dash, have this feature explicitly built-in.

JAXmag: According to the study, by November 2013, the amount of inflows attributable to "sin" entities had shrunk to maximum three percent of total transactions. Why is that? Was this the moment when it became obvious that Bitcoin could eventually go mainstream?

Tasca: As we mentioned in the paper, the takedown of the Silk Road was a big wake up call for darknet users who wrongly assumed that bitcoin was 100 percent anonymous. This single site also accounted for the lion's share of illegal business at its height. At the same time, you saw entrepreneurs from the technology and financial sectors swoop in backed by VC money. I don't think Bitcoin is mainstream yet, but it's on the right path to get there eventually.

JAXmag: Is Bitcoin a safe choice?

Tasca: Bitcoin is as safe as you can get thanks to the blockchain, encryption, and the vast mining network that validates all transactions. But it's also vulnerable to being lost (if you lose your private keys), but that's the same as if you have cash in your wallet and you lose it. Thanks to some innovative start-ups, web- and cloud-based services to securely manage those private keys is making that less of a problem.

JAXmag: Will blockchain outshine Bitcoin? If yes, where do you expect to see more blockchain activity?

Tasca: Although bitcoin is certainly the most famous, widely used and successful current application of blockchain, it is destined to recede in favor of other blockchain applications. On this regards, I am agnostic with respect to debate around token-based or token-less blockchains and the debate around public or private ledgers. It really depends on the business cases. I honestly think that some blockchains will serve better their goals if implemented with a native token and kept open. Similarly, I think that for good confidentiality and cost efficiency reasons, some blockchains will serve better their goals if implemented without a native token and if kept private.

Our socio-economic systems will be probably adopt different blockchain infrastructures which will coexist together.

It is difficult to guess what will be the blockchain killer app in the year to come because blockchain technologies can find application in different domains. During the last two years, most of the interest and activities came from the financial industry. For the next years, I expect other sectors like energy, telcos, media and healthcare to be more active. New opportunities will emerge for them to redesign their business logic.

©istockphoto.com/Inok

Blockchain will not eliminate the middleman

Looking beyond the blockchain hype

Blockchain comes in handy if you don't want to put your trust in a specific third party. It is great if you want the transactions made in the system to be unchangeable but it also adds inefficiency compared to regular databases. It can be slow, cumbersome and it has scaling problems. Blockchain is not perfect.

by Henry Brade

Blockchain is a relatively new concept that was first introduced in the Bitcoin whitepaper in 2008 as a "chain of blocks". It is a critical component of Bitcoin, the digital currency that allows people to send and store money without the need to trust a specific third party. It is a fully transparent decentralized ledger shared by a peer to peer network. It has not just removed the need for trust, it has created a new type of accounting system that is immune to changes or counterfeiting.

Now there is great hype about this blockchain. More and more money is poured to R&D in order to figure out what else can be done with a blockchain besides the management of the Bitcoin system. In theory many things could be stored in a blockchain based ledger but is it actually a good idea? Not necessarily.

Blockchain is great if you do not want to trust a specific third party. It is great if you want the transactions made in the system to be unchangeable. It is great if you want a high level of auditability. If you want speed, efficiency, scalability – then blockchain is not necessarily the best option. Instead, it could be useful to look at regular database solutions.

Blockchain is not a magic bullet

Then there is the problem of data that is blockchain native versus data which is not. As long as all the data in the blockchain is created and stored in the blockchain and can be cryptographically verified, the system can achieve a fully trustless status. This applies to Bitcoin, as this currency only exists in

the Bitcoin blockchain. If you try to store data in the blockchain that is created elsewhere, we circle back to the trust issue.

For example, if I include status information from IoT sensors to blockchain, who or what guarantees that the data from the IoT sensors is accurate and not faked? The blockchain cannot guarantee that; it can only guarantee data that exists in the blockchain itself. This does not mean blockchains are useless for everything else. It only means that blockchain in another type of scenario is less useful than with Bitcoin. In other scenarios, it does not fully eliminate the trust issues, however it can still be of use.

Bitcoin's blockchain

In 2016, one million transactions were recorded to the Bitcoin ledger that included extra data as part of the transaction. This is a significant increase from 2015. Many of these transactions were likely related to timestamping data that exists outside the blockchain. Blockchain timestamps are useful since the provide irrevocable proof that the timestamped data existed at a certain time.

The Bitcoin blockchain is the most secure platform for the purpose of trusted timestamping. It is secured by the most powerful computer network in the world, it is highly reliable and extremely robust. Bitcoin is running 24/7/365. The last time the Bitcoin network experienced an outage was in early 2013. That problem was solved in a few hours. This level of reliability cannot be provided by any other blockchain platform at the moment.

Problems blockchain does not solve

It is curious to me that blockchain is often considered a solution to increase efficiency. That is rarely the case. Blockchain often adds inefficiency compared to regular databases. It can be slow, cumbersome and it has scaling problems. What it does offer is the elimination of a trusted third party, increased security and reliability, and a high level of auditability.

In the case of Bitcoin, we have a system that actually takes advantage of all the true benefits of blockchain. Thanks to blockchain, Bitcoin has a currency system with a fixed amount of units that cannot be manipulated or changed by any human being. It has a transaction system which allows for completely irreversible transactions. It allows for accounts that cannot be frozen by anyone in the world. And, at the same time, it allows value transfer across the world with little friction.

Bitcoin is a true revolution of the whole financial system. It is the first asset that is fully independent, cannot be manipulated, cannot be frozen, cannot be controlled. It allows people to actually take control of their own money instead of being forced to rely on a third party such as banks.

Conclusion

Finally, I would like to stress that blockchain is not an invention that will eliminate the middleman. Many services that banks or payment processors now offer are also very relevant in the Bitcoin world. This leads to massive business opportunities. The key difference is that the core system does not force people to use specific third parties. It is their choice to use third parties, if their services offer added value that makes it worth it.

With Bitcoin we are eliminating the natural monopoly of banks and that is a true blockchain revolution. This level of disruption is what I find most interesting and I look forward to seeing more of it in the blockchain space.

"Blockchain often adds inefficiency compared to regular databases. It can be slow, cumbersome and it has scaling problems – but it does offer increased security and reliability."

Henry Brade is at the helm of Prasos Oy, a growing Finnish Bitcoin start-up, one of the first Bitcoin companies in the world. He has been in charge of Prasos Oy's operations since March 2012. The company's product line currently consists of four services: Coinmotion, Denarium, Bittiraha.fi and Bittimaatti. Coinmotion is a leading European bitcoin exchange service. Denarium is producing the next generation physical bitcoins. Bittiraha.fi is the market leader for Bitcoin services in Finland. Bittimaatti is the largest Bitcoin ATM network in the Nordics.

Jump on the Java 8 bandwagon

The use of Java in finance

The fact that Java is one of the hottest programming languages in finance is no secret. However, making the shift to Java 8 might not be so easy, especially since coding in Java 8 is different to coding in previous versions of Java. Still, the advantages outweigh the disadvantages.

by Stephen Colebourne

The Java platform has always been widely used in finance, but there is no doubt that Java SE 8 has helped reinvigorate the platform. Through blogs and conference talks, I've been trying to help developers make the shift to Java 8, emphasising that coding in Java 8 is different to coding in previous versions of Java. Coding standards need to be reconsidered and design patterns altered. But that change is very much worth it.

Java 8 versus previous versions

The most visible change was the introduction of lambdas and the associated streams API. They have definitely changed the feel of the language, allowing code to be written in a much more functional style. Abstractions that used to be verbose can now be done with ease, and this changes the way code is written and designed. But like all new features, there is a balance to be struck. Not all code reads better or performs as well when written using the streams API and lambdas. But where it does work well, it can result in much clearer code.

When lambdas were added, an associated change was to allow methods in interfaces. Thus an interface can now have three kinds of method – abstract, default and static. Where lambdas change code at the micro-level, methods on interfaces change code at the macro-level. New designs for subsystems become possible, and it can take a while to understand that potential. As an example, one pattern I have used frequently is to have an interface with one or more package-scoped implementations. Previously, this would have needed a factory utility class to create the instances, but now a static method on the interface can do the job. This can greatly reduce the amount of public code in an API if used correctly.

java.time.*

Another feature of Java 8 is the new date and time library java.time.* (JSR-310). As the co-spec lead and primary author, I based the design on the principles of Joda-Time, the de facto standard library for versions of Java prior to 8. My goal with java.time.* was to make Joda-Time obsolete. Developers migrating their code to Java 8 should adopt java.time.* classes and remove Joda-Time where possible.

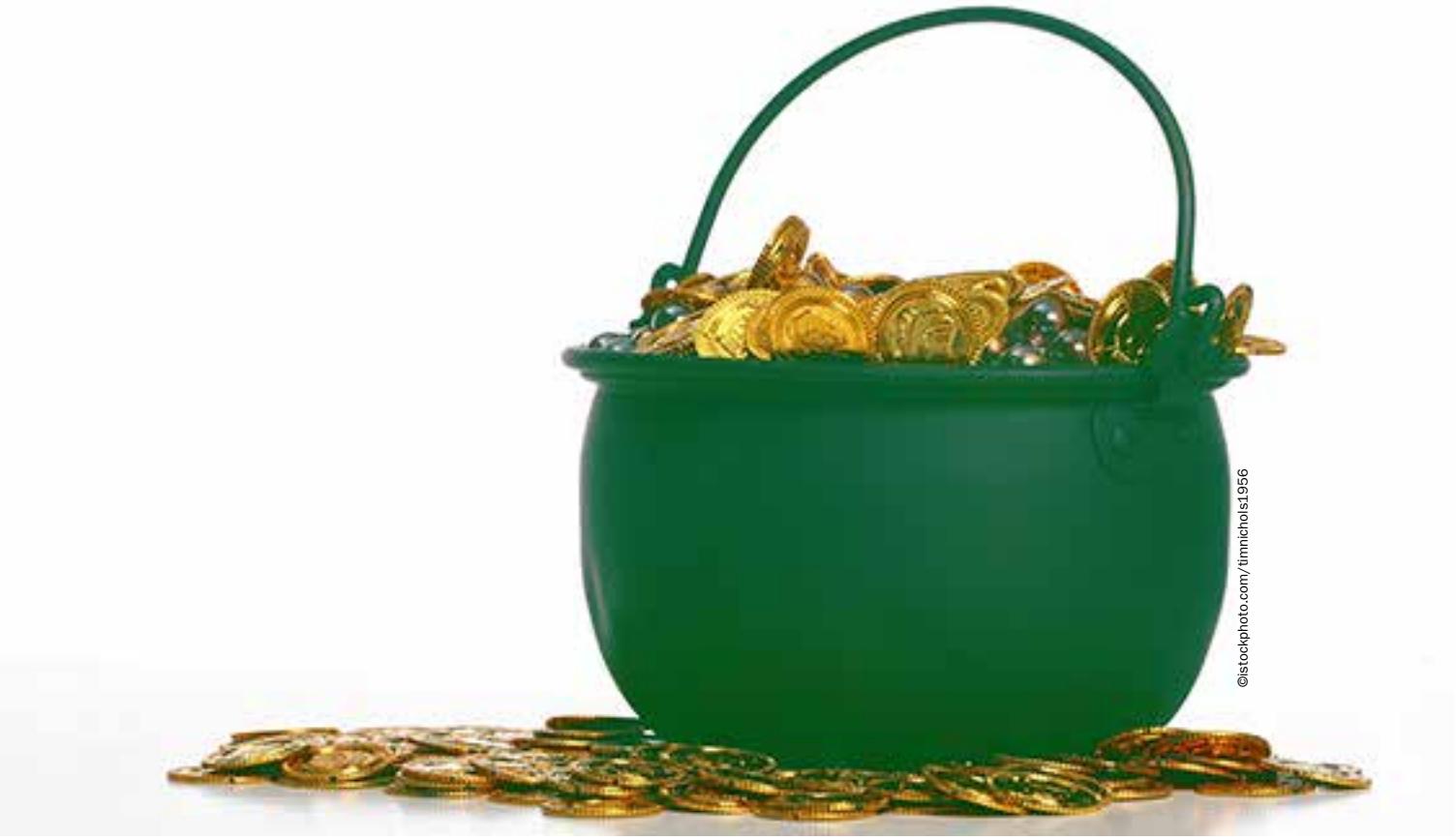
The great part about the new java.time.* classes is that they map so well onto the concepts found in finance. *The LocalDate* class represents a date without time or time-zone, and as such is a perfect fit for many business concepts, such as trade date, effective date and settlement date. Whereas *LocalTime*, a time without a date or time-zone, is a perfect fit to describe when the LIBOR fixing takes place, 11:00. And *Instant* is ideal for collecting a timestamp in monitoring and performance code. The key point here is that Java 8 has made finance in Java much easier – no need for the Joda-Time library any more.

Java in finance

To take advantage of Java 8 and the increasing use of Java in finance, I have spent the past few years at my day job building an open source library for market risk in finance. Strata, from OpenGamma, is a standard Java library available in Maven Central that allows anyone to create and manage financial instruments for the purpose of pricing and analytics. It has the potential to be a key building block for many organizations in the world of finance, with its open source nature ideal for the standardized products that the industry trades since the 2008 crisis. Strata has been written in Java 8 from the ground up, and makes full use of all the new features, with a strong focus on immutability and functional design.

Java SE 9 is on the horizon with its focus on modularization. This will be another big change to Java for developers to learn.

Stephen Colebourne has worked with Java since v1.0. He is a Java Champion and JavaOne Rock Star speaker. At OpenGamma, he produces open source software for the finance industry, but is best known for his work on date and time in Java.



The world is full of patterns – this is one of them

The Leprechaun Trap

Our computing world has been blessed with a lot of wonderful patterns. In this article, Ram Lakshmanan talks about the Leprechaun Trap.

by Ram Lakshmanan

Starting from the fine grain of salt to complex cosmos, there are patterns everywhere. A right pattern withstands the test of time. Patterns are the constant factor in the ever-changing world. It requires a lot of hard labor, rich experience (both good and bad), laser focus and perseverance to crystalize and create patterns. Fortunately, our computing world has been blessed to have such wonderful patterns created. One classic example is represented by software design patterns – Singleton, Factory, Visitor, Observer, Memento created by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. These four engineers have encapsulated their years of learned lessons, refined them and passed them on to us. These patterns are universally applicable, irrespective of what programming language you use (Java, C, C++, PHP, Ruby, etc.), irrespective of what technology stack you run (JEE, .NET, LAMP, etc.), irrespective of what type of application you build (Mobile, Web, SOA, Microservices, Batch, etc.).

Thread dump analysis patterns

Inspired by those four great engineers, I have crystalized, refined and turned the years of production battle fought experience into thread dump analysis patterns. Thread dumps are a vital artifact to do RCA (Root Cause Analysis). Did your application become unresponsive all of a sudden? Did your application's CPU start to spike up without increase

in traffic, without making any code changes or any environmental changes? Did your application's response start to degrade? Did your application start to experience memory problems after running for multiple days/weeks? Answers to several such complex problems are present in the thread dumps. But they are buried inside details. To shed light on those hidden answers, I have created thread dump analysis patterns.

Allow me to introduce you to the "Leprechaun trap" pattern. Objects that have *finalize()* method are treated differently during the garbage collection process than the ones which don't have them. During the garbage collection phase, objects with *finalize()* aren't immediately evicted from the memory. Instead, those objects are added to an internal queue of java.lang.ref.Finalizer object. There is a low priority JVM thread named "Finalizer" that executes *finalize()* method of each object in the queue. Only after the execution of *finalize()* method, objects becomes eligible for Garbage Collection. Because of poor implementation of *finalize()* method if Finalizer thread gets blocked then it will have a severe detrimental cascading effect on the JVM.

If Finalizer thread gets blocked, internal queue of java.lang.ref.Finalize will start to grow. It would cause JVM's memory consumption to grow rapidly. It would result in OutOfMemory Error, jeopardizing the entire JVM's availability. Therefore, when analyzing thread dumps, it's highly recommended to study the stack trace of Finalizer thread.

Real-world example

Here is a sample stack trace of a Finalizer thread which got blocked in a finalize() method:

The above-mentioned stack trace was captured from a JVM which was using one of the older versions of JTDS JDBC Driver. Apparently, this version of driver had an issue; you can see *finalize()* method in the *net.sourceforge.jtds.jdbc.JtdsStatement* object calling *JtdsConnection#releaseTds()* method. Apparently, this method got blocked and never returned. Therefore, Finalizer thread got stuck indefinitely in the *JtdsConnection#releaseTds()* method.

As a result, Finalizer thread wasn't able to work on the other objects that had *finalize()* method. Due to that, the application started to suffer from OutOfMemoryError. In the latest version of JTDS JDBC Driver, this issue was fixed. However, be very careful when you implement *finalize()* method.

Listing 1

```
"Finalizer" daemon prio=10 tid=0x00007fb2dc32b000 nid=0x7a21 waiting for
                                                monitor entry [0x00007fb2cdcb6000]
   java.lang.Thread.State: BLOCKED (on object monitor)
   at net.sourceforge.jtds.jdbc.JtdsConnection.releaseTds(JtdsConnection.
                                                                    java:2024)
   - waiting to lock 0x00000007d50d98f0 (a net.sourceforge.jtds.jdbc.
                                                                JtdsConnection)
   at net.sourceforge.jtds.jdbc.JtdsStatement.close(JtdsStatement.java:972)
   at net.sourceforge.jtds.jdbc.JtdsStatement.finalize(JtdsStatement.java:219)
   at java.lang.ref.Finalizer.invokeFinalizeMethod(Native Method)
   at java.lang.ref.Finalizer.runFinalizer(Finalizer.java:101)
   at java.lang.ref.Finalizer.access$100(Finalizer.java:32)
   at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:178)
```

> *"A right pattern withstands the test of time. Patterns are the constant factor in the ever-changing workld. Fortunately, our computing world has been blessed to have such patterns created."*

Why name it Leprechaun Trap?

Kids in Western countries build the Leprechaun Trap as part of St. Patrick's day celebration. The leprechaun is a fairy character, basically a tiny old man, wearing a green coat and hat in search for gold coins. Kids build creative traps for the leprechaun, luring him with gold coins. Similarly, anxious Finalizer thread is always in search of objects that has *finalize()* method to execute them. In case if *finalize()* method is wrongly implemented, it can trap the Finalizer thread. Because of this similarity, we have chosen to name it the Leprechaun Trap. You can learn about other patterns here: https://blog.fastthread.io/category/thread-dump-patterns/.

Ram Lakshmanan is the founder of the highly popular GCEasy.io – Universal Garbage Collection log analyzer and FastThread.io – Java thread dump analyzer. Ram advices startups to Fortune 500 enterprises to Governmental organizations on their critical technology initiatives. He is the recipient of popular developer contest awards & highly sought speaker in major developer conferences.

Major SaaS data security concerns

What can you do to minimize the risk?

More and more businesses are now opting for Software as a Service (SaaS) instead of on-premise software options. Still, SaaS and cloud data storage are both relatively young technologies and, as with all things new, they both carry certain risks.

by Jorge Sagastume

Knowing what you're dealing with is crucial for successfully running a business. So what are the risks? Can you predict them and what you can do to prevent them? This article will provide some comprehensive answers to those questions. What are the risks and issues of using SaaS? The first rule of crisis management is to anticipate any possible mishaps. So let's start by defining what might go wrong when using SaaS. There are three main concerns when using Saas:

- Data breach – Although this issue is possible even if you use on-premise software, illegal data intrusions as a result of hacking or industrial espionage are a real concern.
- Availability of the software – Internet outage or unscheduled maintenance may make the used software unavailable. With SaaS, you are at the mercy of your service provider and service degradation or inadequate uptime performance may occur.
- Financial stability of your cloud provider – Unfortunately, you have no way of predicting how your cloud provider will do business-wise. If the company goes under, there is a chance you'll be unable to access and process your data.

So what can be done?

It's important to point out that SaaS and cloud providers are continuously working on developing solutions and upgrading their service. Also, users are becoming more informed and proactive. A recent study shows that almost 80 percent of US-based users and more than 70 percent of UK users "are aware of standalone services that allow them to backup and restore SaaS information separate of the provider." This is a clear indication that people are doing what they can to secure their work and data, and not leave things to chance.

With the huge benefits of SaaS services, it would be illogical to avoid using it because of a few minor concerns. Just for illustration purposes, an average cost reduction of IT service expenses when using SaaS services is 15 percent.

So how to minimize the risk and secure your work? To decrease the possibility of a data breach, first of all, you should make sure your SaaS provider has a solid security plan. When negotiating your contract, ask them if they've done a risk analysis and what are their security and disaster recovery methods. Also, implementing your own backup system, separate of the provider, is always a good idea.

For business continuity protection, utilize a software escrow. If your SaaS provider has a severe service interruption, software escrow will enable you to access the software you're using along with your data. It's a great way to ensure that your business can continue, no matter how long it takes to fix the problem with the SaaS service provider.

When dealing with software escrow, you have to make sure that you are protected. Be mindful of the release terms of your escrow agreement and confirm that any software updates are escrow protected.

Although an escrow agreement will also guard you against potential service provider insolvencies, there is a downside to it. If that happens, you will need time to make the necessary infrastructure, like servers or a substitute SaaS solution, operational. Depending on your escrow agreement, this time can be as short as 24 hours. Still, it would be smart to check out the provider's financial and legal standing before you commit to a deal with them.

Conclusion

Like with most things in life, you'll have to be smart when dealing with your SaaS! Keeping in mind all the things which can go wrong and being prepared will make your business operation endeavors relatively painless. Worst case scenarios are quite uncommon, and with any luck, you'll avoid them completely. However, by following our simple instructions you won't be caught off guard if the disaster does strike.

Jorge Sagastume is a Vice President at EscrowTech International, Inc. with twelve years of experience protecting IP and earning the trust of the greatest companies in the world. Jorge has been invited to speak on IP issues by foreign governments and international agencies.

LinkedIn: https://www.linkedin.com/in/jorgesagastume

Choose software re-engineering

Don't fall into the "re-write from scratch" trap

The pace of change in technology can make it hard for businesses to keep up but the need to adapt is anything but a fad. Here's how you can avoid falling into the "rewrite from scratch" trap.

by Sahil Patel

You may feel like you need to adjust your software product or application right after you've built it to make sure it stays relevant and interesting. This adaptation is not just a nice thing to have, it is a necessity. Users will not tolerate poor performance, clumsy navigation or other signs of aging software.

But IT teams often cannot handle upgrades to UX or software reengineering, in spite of the fact that these skills are critical if the company is to keep costs down, keep the project on budget and keep the company apps and software up to date to meet the expectations of users and customers. When it comes to these types of projects, it is important to have expert advice and help or you can easily make the wrong step that could have dire repercussions. There are several factors that can help you make the case for software re-engineering:

- Improved reliability
- Enhanced features and functionality
- Improved user experience
- Improved scalability
- Improved maintenance and support
- Improved ROI and TCO

"Software re-engineering is like the renovation of a house. The designers do not have the luxury to start from scratch – they must work with the existing foundation."

The enterprise may focus on the expense of software re-engineering but there is often little focus on the expense of maintaining the status quo. Status quo may mean the loss of users, increased abandonment on a site or within an app, resulting in lost sales, the need to increase IT support staff, the loss of support or increased cost of support for older technology platforms and products.

In some businesses, software re-engineering is seen as a luxury. In fact, even the smallest local business needs an affordable, dependable technology outreach. The numbers tell the tale. Today, nearly every one of your customers has a smartphone (maybe more than one), a tablet, and perhaps a laptop or desktop computer. Every consumer is familiar with and expects the best support when they use an app or a website and those businesses that do not satisfy these expectations will lose both revenue and customers.

To compete in your market (actually in any market, really), you must stay abreast of the trends, technologies and expected user experience. Software reengineering allows the business to take a step back and look at existing and new ideas for customer, partner and supplier outreach and consider new designs and models to satisfy the market and gain a competitive advantage.

While many enterprises struggle to keep their IT staff abreast of the newest approaches and technologies, there are plenty of IT experts out there who can offer a helping hand. A good software re-engineering project must be managed by a team with comprehensive understanding of trends, technologies, user expectations, integration, networking, configuration and every other aspect of application development.

Software re-engineering is like the renovation of a house

But, beyond these skills and knowledge, it is important to distinguish between the standard application development project and a software reengineering project. Software re-engineering is rather like the renovation of a house. The designers do not have the luxury to start from scratch. They must work with the existing foundation and add technology and features that are compatible with the existing foundation and will allow the business the flexibility to grow and change.

The reengineered solution may add new features and remove redundant or old features or functionality in order to improve the workflow, and enhance the user experience and performance and scalability of the solution.

Software development must include requirements planning, while software re-engineering must include requirements planning and a complete understanding and detailed documentation of the existing software. This crucial step ensures that the project will not be derailed or stalled by unanticipated issues. In the software re-engineering world, one does not want to hear the phrase, "let's go back to the drawing board".

While the management team may believe that a software re-engineering project will be expensive and time-consuming, the cost of starting from scratch or of doing nothing may be extremely high. Lost customers, low user adoption, inefficient solutions, poor performance, and expensive maintenance and support will nag the enterprise. It is better to be proactive and engage an expert software re-engineering team; one that can provide a swift, efficient process that will result in better ROI, TCO, customer satisfaction and performance and provide a competitive edge in the market.

Sahil Patel is an engineer and project manager at Elegant MicroWeb, involved with project management and services delivery. An excellent problem solver with architectural and hands on programming skills on Microsoft platforms, Linux, PHP frameworks, Content Management Systems, mobile and ecommerce frameworks. A team builder with more than twelve years of experience, who inspires and motivates project team to perform consistently that, can meet client's expectations with proven offshore delivery model.

Interview with Doug Schaefer

Eclipse Two: "I want to use Electron as a vehicle to re-think the IDE experience."

Doug Schaefer, Eclipse CDT project co-lead announced in a blog post that he is working on Eclipse Two, "the real next-generation Eclipse IDE based on Electron." The announcement sparked a lot of questions, so we decided to invite him to talk about Eclipse Two and what this project means in relation to the "good old" Eclipse IDE.

According to the Eclipse Two's description on GitHub, "the philosophy is to treat the IDE as a web-site which has access to local resources and tools as well as cloud based services. It brings a new user experience to the desktop IDE that is friendly, integrated, and powerful."

JAX Magazine: You are an Eclipse community veteran and project lead of the C/C++ Development Tools (CDT). From the perspective of a C/C++ developer, what advantages does the Eclipse IDE have over other integrated development environments?

Doug Schaefer: Other than the great features that we offer in the CDT that other IDEs try to keep up with, such as rich content assist and source navigation, as well as a great debugging experience, it really is the ecosystem of plugins that Eclipse offers that makes Eclipse such a popular IDE with C/C++ developers. They need good source management integration and plug-ins like EGit and the Subversion plug-ins offer that. They are often working with other languages, especially as they venture into the world of the Internet of Things. Being able to develop both the server side and the embedded side of an IoT system using different languages but the same tools allows developers to work on the full stack much more easily.

Being able to augment their systems with domain specific languages, it is valuable to be able to have full IDE capability thanks to Xtext. And the systems that C/C++ developers work with tend to be very complex which leads them to be one of the first communities to adopt modeling and Eclipse has such rich modeling plug-ins. And, of course, not to mention the ability for C/C++ developers to build their own plug-ins to customize their IDE and provide features specific to their environments, that would be very difficult with other IDEs, especially those that don't publish their APIs.

Having this rich ecosystem comes from having a strong open community that isn't under any single vendor control making it free for anyone to contribute and extend.

JAXmag: In your blog post, you announced that you are actively working on "Eclipse Two"; its goal was communicated quite clearly – to become the official successor of the "classic" Eclipse IDE. Why do you think this step is necessary?

Schaefer: Well, the goal is really to explore a possible successor. The community will decide whether this new work is worthy of being that successor.

Eclipse Two started out of a thought exercise where I tried to see where Eclipse would be in five or ten years. I think we are severely handcuffed by SWT if we want to be able to provide rich visualizations. I had hopes for JavaFX, but I haven't seen the Java community galvanize around it as a desktop GUI framework of the future. And that led me to wonder about the future of Java as a desktop application environment and whether other technologies might offer us that.

And that led me to Electron, which is also used by the Visual Studio Code and Atom editors. I want to see if we could build a full IDE based on the Chromium browser and node.js that make up Electron. And while doing so, I want to use it as a vehicle to rethink the IDE experience and grow past the 1990's style desktop paradigms that are the foundations of the Eclipse "classic" IDE. It's really fun to think of what you would do if you could do it all over again, and this is the opportunity to do that.

Portrait

Doug Schaefer is the Eclipse CDT project co-lead and a Software Architect at QNX, a BlackBerry company, working on the Momentics IDE.

JAXmag: What are the advantages of building Eclipse as a web IDE, based on technologies like Chromium and node.js (Electron)?
Schaefer: Actually, I don't consider Eclipse Two to be a web IDE. My focus is just on using the HTML DOM APIs as a GUI framework for a desktop IDE. Electron is very good at that. It offers a well-defined browser environment since it includes the browser. It also offers native environment integration thanks to the node.js APIs. It allows us to leverage the complete catalog of libraries that are on npm. It really gives us everything we need to provide a modern user experience while allowing us to integrate with the tools and service developers expect.

What's especially exciting about this direction is the community of developers familiar with these technologies is huge. Our biggest challenge with the "classic" Eclipse IDE is finding people who know Java and are proficient with Eclipse's workbench architecture. And that is especially true in the C/C++ community. Web technologies are prolific, and you even see them in the embedded products with HTML-driven displays or browser based visualization of embedded systems. To have such a large community to draw contributions from will breathe new life into the Eclipse community.

JAXmag: The Language Server Protocol (LSP) will be used to make Eclipse Two polyglot. In your opinion, is the LSP the future of language support in all the development environments or are there other competitive technologies?
Schaefer: The Language Server Protocol effort is the first one I've seen that truly reaches out to all developers to participate. Microsoft is very open to adding extensions to the protocol as different language server providers and client front ends saw the need and that has really made us interested in it from the Eclipse CDT side of things. That openness will foster an ecosystem of language servers that we will be able to leverage, not only with Eclipse Two but with the Eclipse "classic" IDE as well.

JAXmag: Your project Eclipse Two makes use of the advertising phrase „The real next-generation Eclipse IDE based on Electron"; this reminds us of Eclipse Che. In which way is Eclipse Two different from comparable IDEs like Eclipse Che or Eclipse Orion that are also making use of a web editor?
Schaefer: The main difference is that Eclipse Two has no server component. Everything is local, just like the Eclipse "classic" IDE. There is no need to rent time from Cloud services providers, or manage your own server, or to install Docker to run your IDE. It's a classic desktop application that happens to be an IDE built using Web front end technologies and JavaScript libraries that access the local system and can also interact with remote servers.

I understand the need for such server based architectures for IDEs. I don't think they'll be mainstream. For one reason or another, most developers feel secure having the source code they're working with on the hard drives of the machines on their desks. I firmly believe that this will go mainstream in the foreseeable feature and we need to make sure we're providing those developers with the best IDE current desktop application technologies allow.

JAXmag: In your blog post, you also talk about your intention to develop Eclipse Two with the Eclipse community. What are the odds of Eclipse Two being accepted as an official Eclipse project and being developed by the community?
Schaefer: If Eclipse Two is to ever release for the general public to use, it has to be done with the help of the Eclipse community. I don't have the resources to do this myself. Neither does my employer. It will become an Eclipse project or I'll stop working on it.

But as of now, I've had a lot of people express interest, ask questions, and provide suggestions for Eclipse Two. I think it's a very exciting technology and fun to work with. I expect Eclipse Two, or at least something very similar, to build up a community quite quickly.

> *"I want to use Electron as a vehicle to rethink the IDE experience."*

JAXmag: The Eclipse IDE is a big platform for a lot of toolings and applications. Will Eclipse Two still be compatible with existing plug-ins and tools?
Schaefer: I think things like the Language Server Protocol offer a path to reuse components of the Eclipse "classic" IDE. We see an effort led by Red Hat on building a Java language server based on JDT, and efforts are underway by the CDT community to build a LSP provider for C/C++. Those will certainly be reused by Eclipse Two. There is a node module that lets you run a JVM in process and provides an interface to it. I can see that being used to provide access to other Eclipse plug-ins. People have expressed interest in using Eclipse RAP as a mechanism to reuse the UI from Eclipse plug-ins.

That being said, it will be difficult to do such integrations cleanly and with a great user experience so I'm not sure how much reuse is practically possible. But I expect the community to surprise me.

JAXmag: How long will it take (approximately) for the new IDE to be in a state that allows developers to use it in their projects?
Schaefer: That really depends on the community. My first phase is to support self-hosted development at which point I'll start setting up the Eclipse project. Then we'll see how much momentum it builds. I started this work looking forward five to ten years. My hope is to have the IDE ready for adoption by the five-year mark. It could happen much earlier if the community makes it happen. For now, we still have many years of life left in the Eclipse "classic" IDE even after this and we will also continue to make sure it works well for the vendors that rely on it and the users who use it daily.

Interview with Davor Bonaci and Jean-Baptiste Onofré

"In a way, Apache Beam is the glue that connects many big data systems together."

Apache Beam has successfully graduated from incubation, becoming a new Top-Level Project at the Apache Software Foundation. We invited the Apache Software Foundation's Davor Bonaci and Jean-Baptiste Onofré to talk about the project's journey to becoming a Top-Level Project and concrete plans for its future.

JAX Magazine: What is the idea behind Apache Beam?
Davor Bonaci and Jean-Baptiste Onofré: Apache Beam is a data processing system that runs at any scale. Its unified programming model and software development kits (SDKs) enable users to define their batch and streaming data processing pipelines. Beam's runners enable users to execute those pipelines on many processing engines, providing portability and future-proofing, as well as avoiding engine, vendor or cloud "lock-in".

The goal of Apache Beam is to raise the level of abstraction further than any existing system – to decouple the user's business logic from the considerations of the underlying engine. This, in turn, enables user's logic to run on any engine.

Well… this has been done before! Two decades ago, modern programming languages were C/C++ but the compiled executable would run on a single operating system only. Then, Java came along – it raised the level of abstraction, introduced byte code, and made the compiled application portable across operating systems. It became successful and closely followed by C#, Python, Scala, and many others. What these systems have achieved in a general sense, Beam aims to achieve in a very narrow domain – we focus on embarrassingly-parallel, distributed data processing only.

JAXmag: Tell us more about what's under this project's hood. How does Beam work?
Bonaci and Onofré: At the top level, Beam offers multiple SDKs. Users leverage those to construct their own data processing pipeline. Beam then takes that pipeline, breaks it down into individual pieces and transforms it into an engine-independent and (mostly) language-independent form. That pipeline representation is then passed onto one of Beam's runners, which further adapts the pipeline for execution on the given processing engine. In a way, Beam is the glue that connects many big data systems together.

JAXmag: Can you describe a typical use case where the benefits of Beam shine through?
Bonaci and Onofré: You can use Apache Beam for any data processing needs, covering anything from a simple batch ETL pipeline to a complex event-time-based streaming pipeline. Some examples include:

- Finding patterns in data
- Analyzing genomes
- Fraud detection
- Real-time gaming
- … really, anything that needs any kind of data processing

Beam's clean abstractions make it easy to develop such data processing pipelines. Even more, executing such pipelines on a local machine, or an on-premise cluster, or in the cloud, is as simple as running one command, without any code changes.

Portrait

Davor Bonaci is Vice-President of Apache Beam and software engineer at Google. He is serving as a chair of the Apache Beam PMC, and working as a Senior Software Engineer at Google on Cloud Dataflow.

@BonaciDavor

Jean-Baptiste Onofré is an Apache Software Foundation member. He is PMC Chair of Apache Karaf, PMC member for Apache ACE, ActiveMQ, Archiva, Aries, Bahir, Brooklyn, Camel, Falcon, Felix, Incubator, jClouds, ServiceMix, Syncope. Jean-Baptiste is also mentor for incubator projects: Beam, Batchee, CarbonData, Guacamole, Lens, Slider.

@jbonofre

> *"At the top level, Apache Beam offers multiple SDKs. Users leverage those to construct their own data processing pipeline."*

JAXmag: Tell us about the history of Beam. How did the project begin?

Bonaci and Onofré: Apache Beam traces its roots back to the original MapReduce system, which Google published in a 2004 paper, and that fundamentally changed the way we do distributed data processing.

At this point early on, things diverge a little bit. Inside Google, engineers kept innovating and refining the core methodology, and those ideas were shared with the wider community in more scientific papers. Outside of Google, the open source community created its own MapReduce implementation in Apache Hadoop. An entire ecosystem developed, which we all know and love, vast majority within the Apache Software Foundation. Over time, amazing innovation happened in both of these sibling branches, with occasional influence of one on the other.

In 2014, Google launched Google Cloud Dataflow, which was based on technology that evolved from MapReduce but included newer ideas like FlumeJava's improved abstractions and MillWheel's focus on streaming and real-time execution. Google Cloud Dataflow included from the start both a new programming model for writing data processing pipelines, as well as a fully managed service for executing them.

In 2016, Google, along with a handful of partners, donated the programming model to the Apache Software Foundation, as the incubating project Apache Beam. Since then, Apache Beam has graduated from incubation, becoming a new top-level project at the foundation.

JAXmag: What are your plans for the project?

Bonaci and Onofré: We are working on improving end-to-end user experience to create truly frictionless portability that "just works". The Java community coined the phrase "write once, run anywhere" but it took them some time for it to truly materialize. Similar considerations may apply to Beam too.

Besides polishing user experience, the next major milestone for the project is the availability of the first release that guarantees backward compatibility, which would make our project ready for enterprise deployments.

In terms of functionality, we plan to keep improving the core model and distill even more complex data processing patterns into simple and portable abstractions. Finally, we are focused on extending our ability to interconnect with additional related systems and user communities. Some examples include:

- New SDKs, such as Python
- New DSLs, such as Scala or SQL
- New IOs, which improve our ability to read/write data from/to multiple storage/messaging systems
- New runners, such as Apache Gearpump (incubating)

With this, Beam will come close to its vision of running any data processing logic on any engine.

Imprint

Publisher
Software & Support Media GmbH

Editorial Office Address
Software & Support Media
Schwedlerstraße 8
60314 Frankfurt, Germany
www.jaxenter.com

Editor in Chief: Sebastian Meyen
Editors: Gabriela Motroc, Hartmut Schlosser
Authors: Henry Brade, Stephen Colebourne, John Davies, Todor Gigilev, Ram Lakshmanan, Sahil Patel, Jorge Sagastume
Interviews: Davor Bonaci, David M. Brear, Eric Horesnyi, Jean-Baptiste Onofré, Doug Schaefer, Paolo Tasca
Copy Editor: Nicole Bechtel, Frauke Pesch
Creative Director: Jens Mainz
Layout: Flora Feher, Dominique Kalbassi

Sales Clerk:
Anika Stock
+49 (0) 69 630089-22
anika.stock@sandsmedia.com