The answer to Java 9 and modularity

We’re just on the edge of blockchain’s potential
Interview with Brian Behlendorf

Modules are very much a feature for the long-term
Interview with Stephen Colebourne

Java 9 is not a Jigsaw release
Interview with Monica Beckwith
The devil is in the keywords

What comes to your mind when you hear the word Java? How about blockchain, DevOps and microservices? If you want to learn about emerging technologies or dive deeper into topics you’re already familiar with (I’m looking at you Java 9), what better way to do that than to go straight to the source?

You’ll learn all about blockchain from Brian Behlendorf, Executive Director of the Hyperledger Project, Georges Saab, chairperson of the OpenJDK governing board and vice president of development for the Java Platform Group at Oracle and Marcus Biel, Software Craftsman, JCP member and CleanCode Evangelist will tell you more about Java 9 and modularity and Abby Kearns, Executive Director of Cloud Foundry Foundation will talk about diversity in tech. Mike Milinkovich, Executive Director of the Eclipse Foundation is here to talk about Eclipse Oxygen and the Java 9 support while JavaOne Rock Stars Monica Beckwith and Stephen Colebourne share their Java 9 experiences with us.

Tired already? We’ve only scratched the surface – this issue contains 14 names, all carefully chosen to make sure you learn from the best.

Have I mentioned that you’ll be seeing some of these people at this year’s JAX London? This is just a sneak peek into the topics that will be discussed at the conference. Suffice it to say there are over 40 international speakers and industry experts eager to talk to you about the hottest topics in programming.

Did I grab your attention? Then my job here is done. Open the magazine and allow the experts to shower you with information.

Gabriela Motroc, Editor

Apache Foundation recommends against using Facebook BSD+patents licensed artifacts

The React.js Licensing issue has come back to haunt us. In a JIRA ticket about patents, Apache Foundation Legal recommended against using React.js and other software using the Facebook BSD+patents license. Chris Mattmann, Principal Data Scientist in the Engineering Administrative Office and the Program Manager of the Open Source Projects and Applications and NSF Offices at the Jet Propulsion Laboratory (JPL) wrote in the ticket that “no new project, sub-project or codebase, which has not used Facebook BSD+patents licensed jars (or similar), are allowed to use them.”

Java 9 fails – courtesy of “WTF, Java 9?!?”

These days, Java 9 developers have their hands full with jigsaw but that doesn’t mean the community is sitting around doing nothing until September. They took Java 9 for a spin and discovered that not all Java 8 code reacts well to Java 9. As Nicolai Parlog, the author of CodeFX and a passionate software developer, said in the description of the http://java9.wtf/ page, the idea is not to play the blame game but to present the things that break in Java 9.

Spring Cloud Function: Spring Boot with “all the benefits of serverless”

Last month, Spring announced their latest project: Spring Cloud Function, which provides a new programming model for Spring Boot applications, abstracting away all of the transport details and infrastructure. Spring Cloud Function allows developers to keep all of their familiar Spring tools and processes, making it easier for them to focus firmly on business logic. Look at it like this – it’s Spring Boot with “all the benefits of serverless.”

IntelliJ IDEA 2017.2: Discover its love for Java 9 and the wealth of new features

IntelliJ IDEA 2017.2 was released in mid-July. This massive update for IntelliJ IDEA contains an abundance of features and important bug fixes. There’s Kotlin 1.1.3 support, Groovy 2.5.0 support, smarter coding assistance, Java 9 module diagrams and more. Let’s put it this way – there are plenty of features that will make you fall in love with it.

Eclipse Oxygen: What’s new, canceled and renewed

It’s been two months since Eclipse Oxygen was unveiled but people are still in awe of the 12th official simultaneous release. This year’s Eclipse Oxygen – the end result of a process that sees the coordination of scheduling and communication across Eclipse Open Source Project teams – includes “the hard work of 83 open source projects, comprising approximately two million net new lines of code.” The output of this process is a composite repository of open source software and a new release of the Eclipse IDE.
Microservices may be a trendy architecture style right now, but they’re a flexible, yet solid foundation for software. In this interview, Kai Tödter describes some of the advantages and disadvantages of microservices. Plus, he explains why cloud computing is a natural support for a microservices architecture.

JAX Magazine: A lot of people jump on the microservices bandwagon without having a clear purpose in mind. How important is it to ask yourself the question: “Should I use microservices?”

Kai Tödter: Often people use software architecture styles just because they are trendy or hyped. So it is very important to ask this question. There are many reasons for using microservice-based architectures but there are always costs that should never be ignored.

JAXmag: How can microservices – if used correctly – offer flexibility in deciding how to best utilize a project’s resources?

Tödter: One of the benefits of a microservice-based architecture is that a single microservice can be totally owned by a small team. So the team members can decide what would be the best technology stack for implementing the microservice. The teams are more flexible and can benefit from existing knowledge and available skills.

JAXmag: What is the correct way to use microservices?

Tödter: There is no general “correct way”, it always depends on the functional and non-functional requirements of the business context. But the teams have to think carefully how microservices should interact with each other, like using orchestration or choreography. The services themselves should be self-contained or use resilience patterns when they need data of other microservices.

JAXmag: Can microservices increase complexity?

Tödter: Yes. While one single microservice might reduce complexity for a specific domain or functionality, the composition of many microservices increases complexity. A few examples are scalable deployment, communication between microservices, centralized logging, monitoring and tracing. And this is just the tip of an iceberg that is often underestimated.

JAXmag: What is the biggest misconception about microservices?

Tödter: I guess one of the biggest misconceptions is that microservices might solve all the existing (architectural) problems. That is definitely not the case, and teams should think...
carefully not only about the benefits but also about the costs when they want to go with a microservice-based approach.

JAXmag: In your view, what are the three golden rules of microservices deployment?
Tödter: I wouldn’t nail it down to “three golden rules”, but there are a few characteristics that should apply to all microservices. For example, each microservice should be independently deployable, upgradable, replaceable and scalable.

JAXmag: What are the best open-source tools for orchestrating microservices?
Tödter: I guess this question is related to container orchestration rather than microservice interaction patterns like orchestration or choreography. Popular container orchestration tools are Kubernetes, Marathon (for Mesos and DC/OS) or Docker Swarm.

JAXmag: What are the key elements in implementing a microservice-based architecture?
Tödter: There are many elements that characterize a microservice-based architecture. I think one key element is that microservices are treated like products rather than projects, meaning that the team owns the whole lifecycle of a microservice, including deployment and operation. A team that owns a microservice should be cross-functional and organized around business capabilities rather than specific technologies (like UI or databases), all skills needed to implement the whole microservice should be in the team. Data management should be decentralized (e.g. each microservice decides his own persistence layer) and technology stacks should be chosen by the team, rather than having a centralized governance.

JAXmag: SAP President Steve Singh said on an episode of CNBC’s “Mad Money” that cloud computing is yesterday’s news – “microservices” are the future [1]. Do you agree?
Tödter: I think Steve used the term “microservice” in a different context. If you think about “cloud computing” as using remote servers hosted on the Internet rather than a local server or a personal computer, then I would agree because microservices are much finer grained on that level.

If you think about cloud computing as using cloud infrastructures, platforms and services, then I would consider the cloud as a great deployment infrastructure for microservices. So, in my point of view, deploying microservices on cloud infrastructures fits very well.

JAXmag: Who should attend your workshop and why?
Tödter: My workshop “Cool Web Apps with Spring Boot, Angular and TypeScript” is for software developers and architects who not only want to get an overview of the used technology stack but also get real hands-on experience building a small microservice from scratch.

References
How to build microservices infrastructure in seven days

It’s possible!

On December 2015, Wix decided to rewrite their aging microservices infrastructure in seven days. Gil Tayar tells their story.

by Gil Tayar

Whenever you ask yourself the question, “should I be using microservices?”, always say yes. Sure, you don’t have to immediately build ten microservices but even if you start with one service that is the whole application, the whole CI/CD pipeline you need to think about and build will easily pay itself back in the long run.

Microservices have most definitely helped us achieved our goals at Wix. The fact that each team can deploy parts of the application without regards to the other team is unbelievably important when we talk about developing with multiple teams. I believe this is the main benefit of microservices: the independence of each team in methodologies, deployment times, tools, and lifecycle.

The biggest misconception about microservices is that it’s about scaling of services. It’s not; that’s just a pleasant side-effect. For most companies, it’s not about the scaling of services, but rather the scaling of the development teams. You can grow your company to hundreds (or as Google has shown, tens of thousands) of developers, where each team is independent of the others.

It’s hard to say what technology is the most important as far as microservices are concerned. What do I love most? The ease of deployment? The orchestrators/schedulers, like Mesos or Kubernetes? The service discovery facilities? The RPC and available eventing mechanisms? Personally, I think containers are the most important tech aspect for microservices. Docker is at the heart of a microservices team’s independence. It has also enabled orchestrators like Mesos and Kubernetes to arise and be mainstream, in turn enabling ease of deployment.

Microservices in seven days

On December 2015, during an internal Wix Hackathon, we decided to rewrite our aging microservices infrastructure. At the time, our microservices infrastructure was slow to deploy and rollback was difficult, and the decision on which servers to run which services was static – you had to figure it out beforehand, and not let the infrastructure handling it automatically. This was great when Wix started using microservices (it was state of the art at the time!) but times have changed.

Should everyone try to match our speed? We did it in seven days, after all. You can too. It’s 2017. It is possible to develop coherent and competent microservices architecture in a short period of time. Maybe not exactly in seven days, like we did. But even a large company with multiple data centers should be able to transition to a microservices architecture with a reasonable amount of speed.

As for how a developer would build a microservices architecture, well, that’s a huge topic. It’s definitely out of scope of my talk for JAX London! If I had to summarize, I would say that building a microservices architecture from scratch is different, and easier, than taking an existing architecture and turning it into microservices.

In our talk, we will see how to build a modern infrastructure that enables you to deploy a self-servicing grid of computers on which microservices can run and discover one another. Will I show the best way to build a microservices infrastructure? No, but I will be showing how easy it is to build using off the shelf components like Mesos, Node, and Nginx. Through describing the process of building one for a Hackathon, you will understand, in a more intuitive way, what consists a Micro-services infrastructure, and what you will need to think about when you will build one.

JAX London “Microservices” track

Interested in learning more about microservices? Gil Tayar will be at JAX London on October 10, 2017. His session, “How to Build a Microservices Infrastructure in 7 Days”, is a part of the microservices track, which contains nine sessions and three workshops. Find out how to break an existing monolith and learn everything there is to know about RESTful Hypermedia APIs and Eclipse MicroProfile. Join us at JAX London!
How do we design software so it doesn’t fail? Are we responsible for reliability? In this interview, Kevlin Henney explains why we all need to work together and reveals why software reliability is more than just a business decision.

What happens when things break? What happens when software fails? We regard it as a normal and personal inconvenience when apps crash or servers become unavailable, but what are the implications beyond the individual user? Is software reliability simply a business decision or does it have economic, social and cultural consequences? What are the moral and practical implications for software developers? And when we talk of “systems”, are we part of the “system”? What about the bugs on our side of the keyboard?

JAX Magazine: What do we mean when we talk about software failures?

Kevlin Henney: We can consider a software failure as an unhandled error condition that upsets not only the software but also something in the world around it, whether that is someone’s temper, someone’s bank account or someone’s social reality.

JAXmag: How can we become more reliable? Should we?

Henney: With every passing moment there is more and more software being developed. There are more people doing it and there are more spaces that software is covering, from in-your-palm apps to industrial control systems, from the convenient to the life critical.

There is an implied responsibility. Is our software reliable enough for such great responsibility? Given the number of error screens in public places that people send me, the number of complaints about time spent – or, rather, lost – in fixing defects and the amount of money businesses hemorrhage through software failures, whether high-profile or not, it seems clear that if reliability is a destination, not only are we not there yet, but we’ve got some way to go.

How do we do it? This is more a question of application than a question of possibility – most reliability issues are solved and solvable problems. Certainly, we don’t know how to address all problems but we know a lot more than we are putting into practice.

So, we take what we know and apply it. We make the business case for it. We make the moral case for it. We embed it in practice and attitude. We make it part of the conversation, part of the culture.

JAXmag: “Move fast and break things” is a famous (and true) motto. Is this a good or a bad thing?

Henney: It is a context-specific thing. When applied within the appropriate context, it can be considered a good thing, an invitation to experiment freely and without restraint, to discover new ways of working and thinking, to break out of an overly comfortable or stuck place. On the other hand, when applied outside the appropriate context it can undermine people and their work, can come across as irresponsible, arrogant and lacking in self-awareness. I would consider this a bad thing.

JAXmag: Is software reliability simply a business decision or does it have other consequences as well? What are the implications for software developers?

Henney: The case for reliability is human and economic. Whether dealing with individuals or companies or society as a whole, a lack of reliability consumes people’s trust, time and money. In the worst cases, it can cost lives and livelihood. In other words, this becomes a moral question.

In some domains the question of reliability threatens to become a legal one, so there is an obvious incentive there. In
other application areas, where users have a choice and also have a desire for convergence of application features and UX, reliability can prove to be a differentiator. Whichever way you look at it – ethics, laws or markets – there is a strong case for increasing reliability.

Development, and therefore developers, need to account for the context in which a component or a software system. What are the implications of failure in that context? Development needs to go further than the technical stack; the full stack includes the world and people around the software.

**JAXmag:** More often than not, architecture is seen as a separate concern from development process. Why is that and what should the relationship between the two be?

**Henney:** Why roles – such as project manager and technical lead – and disciplines – such as development process and architecture – that are ultimately focused on the same thing – in this case, software and its development – end up separating has a long history with many causes and reinforcements. There is a long history of believing that technical work is of lower status than managerial work, which leads to hierarchy and a vertical separation. There is also a long history of horizontal role specialisation in both modern business and in the constantly expanding world of technology – the more there is, the more you need to know, the more you need to develop expertise or draw on expertise, and so on.

It’s true that you can’t know everything or be equally good at everything you do but it also turns out that the role overspecialisation and separation brings a narrowness to development work that is itself a problem. We find many breakthroughs in science, technology and the arts come from synthesis and crossover, from breaking down silo walls and glass ceilings, from walking across strict separations.

The idea that such separation promotes expertise through focus ignores the fundamental communication overheads, mismatched frames of knowledge and practical challenges that come with separating two entangled points of view on the same thing. Separating how you build from what you build is a naive way to approach building, and yet such a separation has captured the imagination – or perhaps lack of imagination? – and orthodoxy of development for too long.

What should the relationship be? It should be intimate. And, like any close relationship, it should attentive, caring and respectful.

**JAXmag:** What’s the biggest error developers make when trying to create a specific enterprise architecture?

**Henney:** That they’re trying to create a specific enterprise architecture. They need to work more loosely, to recognise that the creation of any successful architecture is a fluid activity involving ongoing changes and emerging understanding of both the problem domain and the solution domain.

**JAXmag:** Could you name three anti-patterns of agile adoption?

**Henney:** Yes. Oh... right:

1. Changing the labels but not the actual roles and practices, e.g., phase becomes iteration or sprint, project manager becomes ScrumMaster, status meeting becomes daily stand-up.
2. Churning out functionality without paying attention to technical and team practices. You don’t go faster just because you put your foot down harder on the accelerator; you also have to remember to release the brake, to have fuel and to be aware of the road and the route.
3. Simple as it is, and as old as it is, missing both the subtle and obvious implications of the values of the Agile Manifesto, such as obsessing over processes and tools at the expense of people, or following a plan – whether schedule or architectural – regardless of contraindicating change or feedback.

**JAXmag:** What is your favorite agile tool and why?

**Henney:** The whiteboard. With the exception of my (often poor) font choice, it’s open to possibilities and participation and does not imply an overly strong sense of commitment, i.e., it doesn’t freeze ideas too early or straitjacket thinking to the limitations of code, editors or other software tools.

**JAXmag:** What can attendees get out of your keynote?

**Henney:** Some good stories, a bigger picture, motivation and a clearer way of reasoning about the relationship between software, its behaviour and its unexpected consequences on the world around it.

---

**Fill in the blanks**

Dev and Ops work best together if ... they are considered together.

The biggest obstacle for DevOps is ... that it is treated as a separate role or activity.

What promotes employee satisfaction is ... a sense of progress, a sense of ownership and a sense that the organization around them cares.

The biggest advantage of autonomously-working teams is ... risk reduction through increased group intelligence.

It is important for a positive company culture to ... recognize that culture is dynamic and its assumptions can become embedded or change subtly over time.

---

**JAX London “Software Architecture & Design” track**

Interested in learning more about software architecture and design? Kevlin Henney will be at JAX London on October 9-10, 2017. His workshop, “Architecture with Agility” and his keynote, “The Error of Our Ways”, are both a part of the software architecture and design track, which contains ten sessions and five workshops. Learn how to handle today’s technology challenges and gain insight into the dos and don’ts, ups and downs of modern software architecture. Join us at JAX London!
How do you measure a successful DevOps adoption? What kinds of metrics do you need to monitor, what benchmarks do you need to meet? We talked with Anders Wallgren, CTO of Electric Cloud, about continuous development, microservices, and how we can be better at DevOps.

**JAX Magazine:** How can one achieve success with DevOps? How important is it to measure and track your DevOps efforts?

**Anders Wallgren:** DevOps is a process that works best when teams work in a continuous improvement mode – the improvement in the quality of the daily work is more important than the absolute quality of the daily work. In any continuous improvement environment, measurement and evaluation of important metrics are important to maintain and improve the quality of the work.

**JAXmag:** What mistakes do teams often make when trying to measure their DevOps performance?

**Wallgren:** There are several pitfalls to avoid when selecting and implementing metrics. It is important to avoid vanity metrics, gameable metrics, and shaming metrics and instead focus on metrics that are objective, actionable, and focus on outcomes.

**JAXmag:** What kind of key performance indicators should teams be on the lookout for?

**Wallgren:** Make sure you look at metrics that cover both efficiency and effectiveness. Efficiency covers things that are related to the cost and effort involved in releasing software. Effectiveness covers how well those efforts satisfy the needs of the customers, including customer satisfaction. It’s also important to pay attention to cultural metrics and employee satisfaction.

**JAXmag:** Let’s say a company decides to adopt DevOps. How can the management get engineers on board?

**Wallgren:** Frankly, it’s often the other way around – grassroots support exists and it’s the company leadership that needs to be brought on board. That said, it’s very important to focus on the actual constraints that limit organizational improvement. It is important not to focus resources on efforts that don’t eliminate the current constraints – that’s wasted effort. Metrics help us focus on where the delays and impediments to better performance are located.

**JAXmag:** In your view, what are the most suitable tools that should be part of a ready-to-use “DevOps kit”?

**Wallgren:** One of the most important aspects, as demonstrated in the 2017 State of DevOps report, is automation. Organizations that continue to rely on manual processes will
continue to suffer the consequences: lack of reliable, repeatable software releases. From a metrics perspective, automation becomes a source of much of (though not all) the data related to software pipeline efficiency and even effectiveness.

**JAXmag: Containerization is here to stay and a lot of people say that it hasn’t peaked yet. Organizations use containers in pre-production on a daily basis but that doesn’t mean everyone is ready to use them. What do you think, will the proliferation of containerization into production accelerate?**

**Wallgren:** We are pretty much at the peak of the containerization hype cycle right now. Adoption will continue to accelerate as the tooling matures and measurable benefits are achieved. In production environments, containerization yields the most benefit in microservices architectures, where there are many small services that, composed together, provide application functionality. It is certainly possible to benefit from containers without loosely coupled architectures, but the benefits will be less apparent.

**JAXmag: A lot of people say that microservices should not be used if the company is not embracing DevOps principles. Do you agree?**

**Wallgren:** I largely agree. One of the few benefits of a monolithic architecture is that they can be less complex to deploy and operate. It’s a lot easier to deploy a single WAR file than it is to coordinate the deployment of 150 WAR files representing 150 different services. To fully take advantage of microservices, an organization must be able to rapidly and reliably set up and destroy environments on the fly, as well as layer the microservices on top of that infrastructure.

In addition, monitoring and logging become more difficult, and also more critical, when running microservices. Organizations starting with a blank sheet of paper should strongly consider microservices. For legacy code, it’s less clear what the correct approach is.

**JAXmag: Is DevOps the key to achieving full agility?**

**Wallgren:** At this point, I see DevOps as a logical extension of Agile (but get ten DevOps/Aglists in a room, and you’ll have at least eleven opinions on this matter). Agile has always been about the delivery of software to customers, but the reality is that many organizations still practice so-called water-scrumfall where the agile portion is practiced only in a portion of the software pipeline, usually by the engineering teams, after which release and deployment efforts still use old techniques. DevOps helps organizations “plumb the last mile” from development all the way into production, as well as working to establish feedback loops back into development efforts.

**JAXmag: What is the biggest misconception about DevOps?**

**Wallgren:** The biggest misconception is that you can buy DevOps in a box. I say this as a platform vendor in the space: you need tooling, but if you think that’s all you need to be successful at DevOps, you will fail, and probably waste a bunch of time and money in the process. DevOps is about process, tooling, and culture. All three legs of that stool need to be addressed to realize the payoff that DevOps demonstrably provides.

**JAXmag: What can attendees get out of your talk?**

**Wallgren:** This talk will be a frank examination of what it takes to use metrics to support DevOps transformations. Attendees will come away with actionable content they can use in their own organizations to implement metrics that are actionable and goal-focused. We will cover all aspects of the software pipeline and the types of metrics that are useful at each stage.

---

**JAX London “DevOps & Continuous Delivery” track**

Interested in learning more about DevOps? Anders Wallgren will be at JAX London on October 11, 2017. His talk, “Measuring DevOps: the Key Metrics that Matter”, is a part of the DevOps & Continuous Delivery track, which contains eight sessions. DevOps’ transformative power derives from the fact that it’s simultaneously occurring on a technological, procedural, and cultural level. Aligning business, development, and operations leads to extraordinary new heights of high performance and agility. Get deep insights in the lessons learned from the pioneers of the DevOps movement! Join us at JAX London!
Implementing a continuous delivery pipeline is not trivial, and the introduction of container technology to the development stack can introduce additional challenges and requirements. We talked to Daniel Bryant, CTO of SpectoLabs, about the challenges, tradeoffs, and impact of bringing together containers and CD.

JAX Magazine: What impact do containers have on CD?
Daniel Bryant: That’s a great question. The top three takeaways from my talk are related exactly to this! From my experience, when we introduce containers into a continuous delivery build pipeline, the container image must become the “single binary”. This is the “unit of deployment” that gets built within the first stage of the pipeline and is the artifact that any tests are executed against. Adding metadata to containers images is vital, but often challenging. We must validate constraints (NFRs) that are introduced by the underlying container infrastructure, such as the effects of running within a container with limited CPU or using a virtual network overlay.

JAX Magazine: What is the difference between containers and virtualization technology?
Daniel Bryant: As the answer to this question could be quite long, I will instead recommend reading the article “Containers vs VMs: Which is better in the data center?” [1].

JAX Magazine: Container technology is hardly new. Still, Docker seems to be the most popular player. Are there other players that caught your attention?
Daniel Bryant: No, you’re right – container technology has been available for quite some time, for example Solaris Zones, FreeBSD Jails and LXC. However, Docker was the first to provide a great developer experience with containers by creating good APIs and a centralized container image registry. They were also the leaders in marketing within this space.

There are plenty of other container technologies: CoreOS’s rkt and Canonical’s LXD are interesting in the Linux container space; Intel’s Clear Containers and Hyper.sh’s runv offer interesting hybrids between containers and VMs. The Docker story is also still evolving, with what the creation of the Moby project and the contribution of containerd to the CNCF.

In reality, this is probably only interesting if you are working in the infrastructure engineering space. Increasingly, we are seeing container implementation details being pushed away from a typical developer’s workflow. This is a good thing in my opinion. In addition, standardization is taking place throughout the technology stack. This will increase interoperability, such as the Open Container Initiative (OCI) container image and runtime specifications, and runc and cri-o.

JAX Magazine: What tradeoffs should we be aware of when using containers?
Daniel Bryant: The core advice is that as containers offer great power, they also demand operational responsibility. This advice especially relates to developers creating images. In my experience, although Docker enables rapid experimentation and deployment, it is often the case that developers don’t have much exposure to operational concerns, such as provisioning infrastructure, hardening operating systems, or ensuring configuration is valid and performant. By packaging application artifacts within a typical container image, you will be exposed to these issues.

Portrait

Daniel Bryant is a software developer and CTO at SpectoLabs. He currently specializes in enabling continuous delivery within organizations through the identification of value streams, creation of build pipelines, and implementation of effective testing strategies. Daniel’s current technical expertise focuses on “DevOps” tooling, cloud/container platforms, and microservice implementations. He also contributes to several open source projects, writes for InfoQ, O’Reilly, and Voxxed, and regularly presents at international conferences such as OSCON, QCon and JavaOne.
For example, I have heard of worst-case scenario where production containers were deployed with an old and unpatched “full-fat” operating system, running an application server in debug mode with a wide range of ports exposed.

**JAXmag: Do virtual machines offer better security than containers?**

**Bryant:** Yes and no! Yes, in that the application artifacts that are now running can be isolated at a more granular level. This can increase the core security principles of “defense in depth” via the appropriate use of network ACLs around each service. Or it can apply the “principle of least privilege” by tailoring kernel security configuration using SELinux or AppArmor for each container.

The answer is also no. As hypervisors provide stronger isolation guarantees closer to the hardware, traditional VMs don’t share the Linux kernel like containers do. (And the shared Kernel is the most common attack vector.) However, in fairness, hypervisors have been around a lot longer than container technology, and vulnerabilities are still occasionally found here.

If people are interested in this space, then I recommend reading an article I wrote that summarizes Aaron Grattafori’s excellent DockerCon 2016 talk on “High Security Microservices” [2]. For operators looking towards the future, researching into Unikernels could also be interesting.

**JAXmag: Are containers revolutionizing the IT infrastructure? How?**

**Bryant:** Containers are part of the current revolutionary cycle, which includes the co-evolution of architecture in microservices, infrastructure like cloud and containers, and practices such as DevOps, continuous delivery and Infrastructure as Code.

**JAXmag: What are your favorite container tools right now?**

**Bryant:** I like a lot of the Cloud Native Computing Foundation (CNCF) technologies [3]. The Governing Board and Technical Oversight Committee are doing great work. For example, Kubernetes for orchestration and Prometheus for monitoring (combined with gRPC and Linkerd) are firm favorites within the industry. Other people I would like to mention include Weaveworks [4], who are producing a lot of great tooling around the cloud native continuous delivery experience; CoreOS, who are innovating with rkt and Kubernetes Operators; and Sysdig [5], who produce some great container debugging tools.

**JAXmag: What are some of the common mistakes when introducing containers to the development stack?**

**Bryant:** These are the ones I have seen (and made!) the most:

- Not deploying to a container enabled/supporting platform
- Investing too much time and resource creating a platform
- Treating containers like VMs
- Packaging full operating systems into containers, rather than using something like Alpine, Container OS or RancherOS
- Packaging a deployment artifact late in the continuous delivery pipeline i.e. not running tests against the containerized artifact
- Not understanding that some technologies don’t play well with container technology e.g. several Linux tools (and the JVM) aren’t fully cgroup-aware, and generating entropy for security operations can be challenging on a host running many containers.

**JAXmag: Have you ever found one area where containers can really make a difference, what would that be?**

**Bryant:** In my opinion, the key advantage with containers that wasn’t quite realized with VMs is the ability to easily package application artifacts in an underlying platform agnostic way. This allows developers to run a more production-like configuration locally. It can also facilitate the transition of artifacts between Dev and Ops.

**JAXmag: What can attendees get out of your talk?**

**Bryant:** Hopefully, the answers above have provided some hints. Other than that, all I say is that if you are looking to implement continuous delivery with containers, then you should definitely join me at JAX London!

**References**


---

**JAX London “DevOps & Continuous Delivery” track**

Interested in learning more about DevOps? Daniel Bryant will be at JAX London on October 10, 2017. His talk, “Continuous Delivery with Containers: The Good, the Bad and the Ugly”, is a part of the DeOps & Continuous Delivery track, which contains eight sessions. DevOps’ transformative power derives from the fact that it’s simultaneously occurring on a technological, procedural, and cultural level. Aligning business, development, and operations leads to extraordinary new heights of high performance and agility. Get deep insights in the lessons learned from the pioneers of the DevOps movement! Join us at JAX London!
No one could have seen blockchain coming. Now that it’s here, blockchain has the potential to completely reinvent the world of financial transactions, as well as other industries. In this interview, we talk to Brian Behlendorf about the past, present, and future of this emerging technology.

**JAX Magazine:** Open source is crucial for the success of a lot of projects. Could you talk about why blockchain needs open collaboration from an engaged community?

**Brian Behlendorf:** I believe we are heading towards a future full of different blockchain ecosystems for different purposes. Many will be public, many private, some unpermissioned, some permissioned – and they’ll differ in their choice of consensus mechanism, smart contract platform, security protocols, and other attributes, and many will talk to each other. To keep this from becoming a confusing mess, or a platform war, collaboration on common software infrastructure is key. The Open Source communities behind Linux, Apache, and other successful platform technologies have demonstrated how to do this successfully.

The beauty of cross-company and industry open source projects is that organizations can share the unprofitable and unsexy work of building the libraries and standards, which underlie systems. A shared code base also serves as an excellent way of concurrently building a standard for coexisting on a blockchain. Hyperledger aims to provide tools for communities to build their own chains, rather than driving everyone to one chain. We envision a world of many chains, some public like the crypto-currencies and some

**JAXmag:** Could you tell us more about Hyperledger?

**Behlendorf:** Blockchain technology has been used in crypto-currency settings for several years now. But there are many more uses for blockchain technology beyond crypto-currency. Any business network that records transactions – and can’t depend on a central proxy that everyone has to trust, and for which those ledgers must be immutable – can be remade through blockchain technology.

Hyperledger is an open consortium bringing together a community of organizations and individual developers to establish protocols and standards to improve the performance and reliability of blockchain technology. Hyperledger aims to provide tools for communities to build their own chains, rather than driving everyone to one chain. We envision a world of many chains, some public like the crypto-currencies and some

---

**Portrait**

Brian Behlendorf is the Executive Director of Hyperledger Project. Behlendorf was a primary developer of the Apache Web server, the most popular web server software on the Internet, and a founding member of the Apache Software Foundation. He has also served on the board of the Mozilla Foundation since 2003 and the Electronic Frontier Foundation since 2013. He was the founding CTO of CollabNet and CTO of the World Economic Forum. Most recently, Behlendorf was a managing director at Mithril Capital Management LLC, a global technology investment firm.
“permissioned” – like you will likely see in healthcare settings. By developing a common distributed ledger technology that is shared, transparent and decentralized, the possibilities are endless.

JAXmag: According to a recent report [1] by the European Parliament, blockchain could change the lives of EU citizens. What’s your take on that?
Behlendorf: In the early 1990’s, there was a fairly compelling sense among those who were building the first web software and websites that this was going to “change everything”, by changing the fundamental cost and process of accessing and sharing information. We could not have specifically predicted Amazon, Google, Facebook, etc., but I think that sense that the laws of physics somehow been altered was clear, and the ramifications would take a long time to play out.
That seems to be the case here – it seems like any industry that touches transactions – e.g., all of them – will be impact ed in one way or another. What this means for consumers is anyone’s guess – will it mean being able to manage your personal healthcare records through a “wallet” of some sort? Possibly. Will it mean inter-bank or even international payments settle immediately rather than over the course of days? Definitely. Will it mean the provision of government services – particularly when it comes to property registry, permits, voting registration, and other process/transactional touch points – will dramatically change? Hopefully! But specifics will prove elusive.

JAXmag: The author of the European Parliament report states that “since other applications can ‘piggyback’ the Bitcoin blockchain, the biggest impacts of Bitcoin may be found outside the currency domain.” Do you agree with this statement?
Behlendorf: Crypto-currencies are one possible application on top of distributed ledgers but not the only ones. At times, having a token generated like a currency will be a useful way to fund the consensus mechanisms, or even provide services from distributed apps on a chain. But it’s not the only way, as that accounting overhead adds costs for features that may not be as relevant in a permissioned chain setting. Some apps and “smart contracts” may be built on the Bitcoin chain, and even more on the Ethereum one, but I suspect the majority of transactions and smart contracts will be implemented on permissioned ledgers. I suspect Bitcoin’s utility as an uncensorable store of value will remain its biggest value.

JAXMag: How important is the law in this context? Does the development of blockchain depend on it?
Behlendorf: Technology does not eliminate geography. In the early days of the Web, we hoped to see it as a sovereign place, where ordinary analog-world laws either didn’t need to apply or could not be applied. Reality set in: turns out us humans are citizens of countries, and we benefit from that association as much as are constrained by it, and technology does not provide an escape plan from that. So laws around money laundering, know-your-customer, securities registration, and all sorts of things are likely to continue to apply.

Turning this around, there is substantial hope that one could use distributed ledger technology to automate many of the business processes that today fall to humans to implement, which introduces the potential for corruption, delay, and error. To do that, some laws many need to be re-formed, some regulations made more test-able by algorithm. I’m encouraged to see so many regulators, sovereign banks, governments, and others working to build up their own competencies in this space so that as industries start to implement their business processes as blockchain applications, the regulatory and statutory hurdles can be addressed.

JAXMag: Will blockchain eliminate the middleman? What are the benefits and/or disadvantages of a world without middlemen?
Behlendorf: Yes, the blockchain has the potential to reinvent any transaction that now requires going through a middleman, including finance, banking, contacts, and retail. Before blockchain, buying and selling required an intermediary, a bank or broker who housed your financial data at their computers. When you transfer funds or make a purchase, a banker connects to the bank’s system to record the change. Blockchain replaces this central system with a decentralized ledger of records. Each record is connected to the one before and the one after it, yielding a traceable history of every transaction. No record can be deleted and no existing records can be altered.
In the future, blockchain’s ability to remove the middleman means it could support “smart contracts” with conditional clauses programmed into the blockchain. This makes the contract self-enforcing, by transferring funds only when the conditions are met. Smart contracts could change entire fields of law. Blockchain wills could automatically take effect when a person dies, transferring inheritances without needing an executor. Replacing legal jargon with blockchain logic would require a different type of corporate lawyer with skills akin to a computer programmer.

A distributed ledger gives us the chance to go into those kinds of business networks. Even here, there is sometimes a role for the middlemen; there may still be a need for a central market actor to set the human-defined, human-enforced legal terms by which the participants on that network work together, as well as to define the specific technology stack those participants use to talk to each other. And maybe that central market actor is needed to play policemen from time to time, as a kind of private market regulator perhaps but not to have to be operationally the hub at the centre of this market place.
JAXmag: Have we fully tapped blockchain’s value? If not, how can we do that?
Behlendorf: I think we’re just on the edge of blockchain’s potential. There is no shortage of big visions for this technology as part of a new Internet technology stack that, if done right, it can have tremendous positive impact for the tech industry and society in general. Hyperledger’s role is to enable whatever happens up that stack by building a great, trustworthy, modular, Apache-licensed and transparently built foundation for the base of that stack. We are the plumbers and if we’re successful everyone gets to move on up the stack to more interesting, profitable, powerful applications. By developing a common distributed ledger technology that is shared, transparent and decentralized, the possibilities are endless.

JAXmag: What can attendees get out of your keynote?
Behlendorf: My talk “Blockchain: Why Open Source and Collaboration are Crucial for Tech” will walk through Hyperledger and why it is aimed at bringing an open dynamic to the blockchain community in order to provide governance so that everyone can spend less time worrying about the plumbing and more time building new blockchain products and services. I believe we are heading towards a future full of different blockchain ecosystems for different purposes. Shared plumbing will be a must, governance will be needed, thus collaboration and open source is key. The Open Source communities behind Linux, Apache, and other successful platform technologies have demonstrated how to do this successfully. And the aim with Hyperledger is to do the same.

References


JAX London “Emerging Technologies” track

Interested in learning more about Blockchain? Brian Behlendorf will be at JAX London on October 10, 2017. His session, “Blockchain: Why Open Source and Collaboration are Crucial for Tech”, is a part of the Emerging Technologies track, where you can get into the latest technologies that are shaping our industry’s future. Emerging Tech Day offers the opportunity to learn more about the principles of Blockchain, get first-hand experience with cutting edge tools and tech, and hear exciting insights about the newest breakthroughs in tech from AI to ML.
Blockchain is a game-changer. But how do we handle its development and growth and how do we go from here? In this interview, Eoin Woods explains who is safe from disruption and who needs to step up their game.

JAX Magazine: According to a recent report [1] by the European Parliament, blockchain could change the lives of EU citizens (and not only). What’s your take on that? What is its potential impact?

Eoin Woods: Blockchain has the potential to bring trust and transparency to many types of record and transaction. This could bring real benefits to public life across the EU and beyond.

JAXmag: The author of the report states that “since other applications can ‘piggyback’ the Bitcoin blockchain, the biggest impacts of Bitcoin may be found outside the currency domain”. Do you agree with this statement?

Woods: Absolutely. We can see this already with systems like Everledger, Genecoin, Provenance and Civic using the Bitcoin blockchain for applications from tracking diamonds to providing secure personal identities, I think this is already happening, and you can see it happening even more on other blockchain platforms like Ethereum that were specifically built as distributed application platforms. It’s hard to know if Bitcoin’s impact will be bigger as a currency or a platform, but its use as a platform is already an established fact.

JAXmag: How important is the law in this context? Does the development of blockchain depend on it?

Woods: So far this hasn’t been a major factor in the development of blockchain but clearly that is changing. Nearly all financial services regulators are investigating blockchain as an enabling technology and as regulators and large enterprises start using it for real, legal underpinnings for blockchain transactions will become important.

JAXmag: What are your concerns with regard to blockchain?

Woods: I have a lot of concerns about it being used for totally unsuitable applications just because people want to use it. We’re seeing this already across a number of industries. I have concerns about whether the technical communities around the major public blockchains like Bitcoin and Ethereum can respond quickly enough to the technical problems that are bound to emerge. That said, both communities have responded effectively to challenges including the Ethereum DAO breach and the Bitcoin scalability concerns.

JAXmag: What is this technology’s main advantage? Could you name some of the obstacles?

Woods: The key feature that blockchain brings to an application is the ability to share a verifiable database reliably between an arbitrary number of parties who don’t trust each other. This is a pretty unique feature. The fault tolerance is another very attractive feature, when the number of nodes gets large.

Some of the challenges with it are the potential latencies in getting updates applied and agreed and the inherent complexity of operating and developing against the platforms. The
relative immaturity of the technology is another short term challenge.

JAXmag: In your view, which industries cannot be disrupted by blockchain?
Woods: That’s a good question. While blockchain can potentially disrupt many situations, most of them are cases where intermediaries are providing little more than record keeping and dispute resolution. There are many situations where this isn’t the problem and so blockchain isn’t going to change anything – providing legal advice might be one example.

Cause for concern?
JAXmag: Did we open Pandora’s box when we allowed blockchain and bitcoin into our lives?
Woods: No more than when we invented the Internet!

JAXmag: Will blockchain eliminate the middleman? What are the benefits and/or disadvantages of a world without middlemen?
Woods: Yes, in general, blockchain allows many “middlemen” to be eliminated. However, those that are eliminated are the middlemen who aren’t adding any inherent value to a transaction and are just keeping records and mediating in the transaction. Where an intermediary is bearing risk or providing value in the transaction they are quite safe – they’ll just be participants in the value chain captured on the blockchain.

JAXmag: How can businesses become successful in a world that’s obsessed with blockchain?
Woods: Most businesses should focus on their clients and make them successful and not worry too much about blockchain! However, those businesses that are inherently transactional intermediaries need to honestly reevaluate their business models and be clear about what value they bring to the interaction. This may involve being ready to reprice their services or even change the business they are in, if they’re not adding enough value to survive the disintermediation that blockchain brings.

Is blockchain “the next internet”?
JAXmag: IMF claimed in its piece called The Internet of Trust [2] that blockchain can become “the next internet”. What’s your take on that?
Woods: One way of looking at blockchain is as a trusted data distribution and sharing mechanism, in which case, you could see it as a “new Internet”. This is how Swarm [3] sees itself being described as “a project towards the ambitious goal of building the third web in the ethersphere”. I think we have a long way to go before we need to worry too much about blockchain replacing the Internet though.

JAXmag: What can the IT world learn from the blockchain technology?
Woods: That it’s sometimes worth rethinking your fundamental assumptions. If Satoshi Nakamoto hadn’t done that, we’d never have had a blockchain.

JAXmag: Have we fully tapped blockchain’s value? If not, how can we do that?
Woods: I don’t think we have even started, we’re still in the early adoption and technology development phase, where we’re learning what the possibilities are. To find the value we need to experiment with it in as many problem domains as possible.

To read the complete interview series with nine blockchain influencers, check out www.JAXenter.com.

References

JAX London “Emerging Technologies” track
Interested in learning more about Blockchain? Eoin Woods will be at JAX London on October 10, 2017. His session, “A Breathless Tour of Blockchain”, is a part of the Emerging Technologies track, where you can get into the latest technologies that are shaping our industry’s future. Emerging Tech Day offers the opportunity to learn more about the principles of Blockchain, get first-hand experience with cutting edge tools and tech, and hear exciting insights about the newest breakthroughs in tech from AI to ML.
The release of Java 9 has been delayed to September 21, 2017. What does this mean for Jigsaw? What does this mean for developers? We talked to Georges Saab, chairperson of the OpenJDK governing board and vice president of development for the Java Platform Group at Oracle about the release of Java 9, the roadmap towards Java 10 and more.

The release of Java 9 has been delayed to September 21, 2017, Mark Reinhold, the Chief Architect of the Java Platform Group at Oracle announced in early June. After the EC’s decision not to approve the Public Review Ballot for JSR 376 (the Reconsideration Ballot was approved) and the series of conference calls, no one can say they haven’t tried to stick to the schedule. However, since moving through the JSP process takes time, Java 9 has been delayed again.

Why “again”, you ask? Because Java 9 already received a four-month extension last year, when Reinhold announced that even though there has been significant progress on Project Jigsaw, “at this point, it’s clear that it needs more time”.

JAX Magazine: Java 9 was delayed again. This time for a different reason, although still related to Jigsaw. What does the delay mean for Jigsaw? It is, after all, the star of the Java 9 release.

Georges Saab: Jigsaw is the key driving force of Java 9 so there is no Java 9 without Jigsaw. However, the delay does not mean there’s a change per se to Jigsaw.

JAXmag: There will definitely be Java 9 features that pose a challenge to developers but is Jigsaw one of them?

Saab: Historically, the main issue we have seen with adoption is when one bit of code ignores the public API of another and instead calls into what is meant to be internal implementation details. This means that if that internal implementation changes, that calling code may break. Over the years we have taken great pains to avoid (even) this kind of change – which taken to its extreme risks having evolution grid to a halt. This is one of the main reasons for the introduction of the module system, which allows encapsulation of internal implementation details and restricts other modules to only call the public API. Once in place, this will make it far easier for the entire ecosystem to evolve rapidly.

To allow migration to occur gradually, we recently proposed that the enforcement of this encapsulation be introduced over time. So it is not the module system itself, but the stricter encapsulation which most pre-existing code is likely to run into.

Of course, it is possible to get started on JDK 9 without modules, the class path continues to work, and this is how many developers will likely get started with JDK 9.

There are a couple of other changes which people may run into, one is the change in the format of the version string (JEP 223) and another is to the change in the packaging of the JDK itself (for instance, there is no more rt.jar).

JAXmag: Project Jigsaw should have been a major feature of Java 8. Why was it postponed to Java 9? Does it also have a central role in Java 10?

Saab: It was postponed to 9 so that we could focus on the key features of Java 8. It plays a central role in the future since it...
provides the basis for more rapid evolution of the platform while ensuring the excellent compatibility for which Java is known.

JAXmag Do you expect the adoption of Java 9 to be swift?  
Saab: JDK 8 was adopted more quickly than any major release in the past. I expect JDK 9 to be slower than 8 but on par with other major releases in the past. Going forward in a modular ecosystem, I expect the adoption of new releases to be far more rapid than they have been in the past.

JAXmag Were there features that you wanted to put into Java 9 but ended up not including them? Will we see them in Java 10 instead?  
Saab: There were dozens of projects which made it into JDK 9 in addition to Jigsaw. Of course, there are a lot more we are working on for the future, and some we felt would benefit from additional time and feedback from people using them – which is one of the reasons we introduced the notion of ‘incubator modules’ in JDK 9 (see JEP 11). As an example, JEP 110 – the HTTP/2 client API is an incubator module in JDK 9.

JAXmag: What’s on the roadmap towards Java 10?  
Saab: In the past, major features have been tied to releases – which has meant that if those features needed longer the release would have to slip. Going forward, features will not be tied to releases in advance – instead releases will happen frequently at fixed times, with features being targeted to a release once they are essentially done. In addition, there is now a way of testing certain kinds of new features.
Java Champion Stephen Colebourne is very excited about the upcoming Java 9 release and the imminent modular future. Project Jigsaw should have been a major feature of Java 8 but now it’s ready for public display. Therefore, it’s time to consider how it might affect existing and new codebases. We talked to Stephen about the new Java 9 and the star of the show, Project Jigsaw.

JAX Magazine: Georges Saab, chairperson of the OpenJDK governing board and vice president of development for the Java Platform Group at Oracle told us in early June that many developers will probably get started on JDK 9 without modules. How do you feel about the modular ecosystem?

Stephen Colebourne: The Java Platform Module System (JPMS) is not perfect, but it has reached a point where it is worth releasing. Most developers can continue to use the classpath, and be unaffected by the module changes. Dodgy reflection practices will be the main issue, with some library upgrades necessary to solve them. I’m happy enough that the module system work is winding down, as I personally think there are some other big language challenges that have been delayed too long.

JAXmag: This major new feature is long overdue but now that it’s finally happening, how will it affect existing and new codebases?

Colebourne: In both cases, there is little a developer can do until their dependencies are at least partially modularized. While automatic modules, in theory, allow use of the module system immediately, in practice it is better to wait until dependencies are modularized. What does matter for new codebases is to have a plan for modularization. Structure code into Maven-style projects and ensure that no package is in two modules. This will prepare the way for adding module-info. java files later.

JAXmag: There will definitely be Java 9 features that pose a challenge to developers but is Jigsaw one of them? What features, if any, do you find particularly challenging?

Colebourne: Applying Jigsaw to a mature codebase will likely be very challenging. It may well be so challenging that it will not be worth doing. Modules are very much a feature for the long-term.

JAXmag: Can you give us some examples of how modules can improve the Java 9 experience?

Colebourne: If you can get to a system where your code is modularized and all your dependencies are modularized, then your code will take a step forward in stability and encapsulation. After 20 years, the JVM will actually understand jar file (module) boundaries and respect/enforce them. This is a kind of “tough love”, but will result in better solutions.

JAXmag: Even though Mark Reinhold said at the last Java-One keynote that Java 9 is much bigger than Jigsaw, this feature is undoubtedly the star of the release. But what about those other JEPs? Which one do you like best?

Portrait

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.
Interview

Colebourne: Private methods on interfaces. The ability to add static and default methods to interfaces has really changed how I design APIs. Having private methods will help that even more. But I do encourage people to always write “public static”, “public abstract” or “public default” on interfaces now to gain clarity.

JAXmag: There's a new project by Nicolai Parlog called WTF, Java 9 where he collects short, self-contained, correct (or not) examples demonstrating what fails (in Java 9) and how to fix the hiccups. What problems have you personally encountered when experimenting with Java 9 and how did you fix the problem?
Colebourne: I tried some very early versions and lots didn't work. I've not encountered any significant problems recently.

JAXmag: What is the most important misconception about Java 9? How about Jigsaw?
Colebourne: Most existing Java code will continue to work on Java 9. The classpath still exists. Modules are opt-in. And in my opinion, most application developers should ignore the module system until their dependencies have been at least partially modularized. So, the key misconception is that everything will suddenly fail, when in fact it won’t be much worse than many other Java version upgrades.

JAXmag: What would you like to see in Java 10?
Colebourne: I hope Java 10 will arrive a lot faster than Java 9. As such, I expect it will contain a lot less changes. What I am looking forward to in a future version (not Java 10) is data classes extending to the use cases that developers use POJOs/beans for today. This would be a huge beneficial change for Java.

JAXmag: What will attendees get out of your talk?
Colebourne: My Java SE 9 Modules talk explains the design of the Module system, and how to modularize your own code, while pointing out the pitfalls. I always find it is the “gotchas” that are the most interesting part to explain, and the most useful to attendees.

JAX London “Java Core & Languages” track
Interested in learning more about Java? Stephen Colebourne will be at JAX London on October 11, 2017. His workshop, “Java SE 9 Modules: an Introduction”, is a part of the Java Core & Languages track, which contains fourteen sessions and one workshop. The goal of this track is to help you enhance your skills – whether it’s about learning new tricks, designing an open source library in Java 8 or rediscovering the joy of being a software engineer. Join us at JAX London!
Java 9 is more than just Project Jigsaw. We talked with Java-One Rock Star Monica Beckwith about her favorite improvements in the upcoming Java 9, as well as about the elephant in the room: Project Jigsaw.

JAX Magazine: Georges Saab, chairperson of the OpenJDK governing board and vice president of development for the Java Platform Group at Oracle told us in early June that many developers will probably get started on JDK 9 without modules. How do you feel about the modular ecosystem?

Monica Beckwith: For the long term, it’s a great boost to the Java Runtime Environment and hence the Java ecosystem. Imagine being able to build your application and its runtime environment in a modular format. Then, your customer can deploy it right off the bat without having to worry about the JDK version or the footprint.

JAXmag: What’s the difference in the performance of contended locks between JDK 8 and 9?

Beckwith: If you are looking for pure numbers, then the impact varies based on the amount of contention. The simplest way to think of it is that prior to JDK 9, all inflated locks (the ones that wouldn’t get optimized away) would take a slow path. The contended lock optimization introduced in JDK 9 introduces a few fast path optimizations for frequent code paths. So, you would find things like optimized notify path that would transfer threads in the wait queue directly to the entry queue and optimized entry for inflated locks.

JAXmag: There will definitely be Java 9 features that pose a challenge to developers but is Jigsaw one of them? What features, if any, do you find particularly challenging?

Beckwith: I think with Jigsaw getting its much-needed support from the IDEs, people would feel less threatened by the changes and would eventually welcome it.

JAXmag: Even though Mark Reinhold said at the last Java-One keynote that Java 9 is much bigger than Jigsaw, this feature is undoubtedly the star of the release. But what about those other JEPs? Which one do you like best?

Portrait

Monica Beckwith is an independent performance consultant and trainer optimizing customer applications and systems running the Java Virtual Machine. She has worked with Java HotSpot VM optimizing the JIT Compiler, the generated code, the JVM heuristics and garbage collection and collectors. Monica is a regular speaker at various conferences and has several published articles on topics including garbage collection, the Java memory model and others. Monica led Oracle’s Garbage First Garbage Collector performance team, and was named a JavaOne Rock Star. Monica also co-authored the “Java Performance Companion” book. Monica was recently considered one of the influential women in Java and Scala.
Beckwith: I think String density is a winner here. My favorite may be the contended locking improvements because that’s the first of its kind. But the String density improvements are the biggest that any release has yet seen. The impact will be felt by almost every user.

JAXmag: What is the most important misconception about Java 9? How about Jigsaw?
Beckwith: The number one misconception is that Java 9 is a Jigsaw release. Nope. It’s not true: Java 9 is much bigger than that. Actually, I am currently writing a book on Java 9. There’s so much in this new release, like its performance, the logging interface, the new microbenchmarking harness that’s now a part of the JDK, the fact that the default collector in Java 9 is G1 GC, reactive programming, JShell, VarHandles, segmented code cache, so many things ... The number one misconception about Jigsaw is that it will break everything :)

JAXmag: What would you like to see in Java 10?
Beckwith: I am most excited about Project Valhalla and the enhancements that it will bring to the Java VM.

JAXmag: What will attendees get out of your talk?
Beckwith: Attendees will learn about locking optimizations in Java. I will also introduce an awesome tool called Oracle Developer Studio Performance Analyzer and work through the analysis of finding out the improvements in contended locking. Also, I will be comparing the changes in Java 9 to Java 8.

JAX London “Java Core & Languages” track
Interested in learning more about Java? Monica Beckwith will be at JAX London on October 10, 2017. Her talk, “Java 9: Beyond Contention!”, is a part of the Java Core & Languages track, which contains fourteen sessions and one workshop. The goal of this track is to help you enhance your skills – whether it’s about learning new tricks, designing an open source library in Java 8 or rediscovering the joy of being a software engineer. Join us at JAX London!
All influential people have something in common: they can spread ideas faster and better than anyone else. We are aware that following those people has a handful of perks, including staying on top of the latest news and trends. Therefore, we decided to concoct a list of Twitter accounts all Java fans should follow. The analysis ranks the top accounts according to their social influence, although interestingly enough, not all household names of Java evangelists are on the list. This year’s list includes nine Java Champions.

**Step 1: Infographic**

**Methodology:** We first generated a list of one thousand Java-related Twitter accounts (including all accounts that contain the keyword Java in their bio or in any of their tweets). To score the account and rank them accordingly, we analyzed their social authority and reach using two key metrics: MozRank and Klout.

- **Moz Social Authority Score:** Social Authority score is composed of:
  - The retweet rate of users’ last few hundred tweets.
  - The recency of those tweets.
  - A retweet-based model trained on user profile data.

- **Klout Score:** Klout uses more than 400 signals from eight different networks to update the Klout Score daily. It’s mainly based on the ratio of reactions a user generates compared to the amount of content he shares.

**Step 2: Interview series with Java influencers**

It’s time to see why they deserve to be called influencers. We asked six questions and we received great answers from 11 Java influencers. You’ll find out what they want to see in Java 10, their views on the biggest misconceptions about Java 9 and more.

First stop: Marcus Biel, a Software Craftsman, JCP member and Clean Code Evangelist.

Who has the most influence on Twitter? The JAX London team has trawled thousands of tech accounts to find 20 Twitter profiles that belong in every Java developer’s Twitter stream.

Check out [www.JAXenter.com](http://www.JAXenter.com) for the entire interview series.

Check out the JAX London blog for the detailed scores.
The Conference for Java & Software Innovation
Oct 9 – 12, 2017 | London

CREATE.
INNOVATE.
CODE.

jaxlondon.com
The Conference for Java & Software Innovation
Oct 9–12, 2017 | London

HIGHLIGHTS

60+ international speakers

40+

EXPO

60+
workshops, sessions & keynotes

40+

POWER WORKSHOPS

October 9–12

EXPO reception with top IT companies

CONFERENCE TRACKS

October 10 & 12

- Big Data & Machine Learning
- Java Core & Languages
- Cloud & Modern Infrastructure
- Microservices
- DevOps & Continuous Delivery
- Software Architecture & Design
- Agile & Communication
- Emerging Technologies

POWER WORKSHOPS

October 9–12

- Architecture with Agility by Kevin Henney
- Running Production Microservices by Russ Miles
- Cool Web Apps with Spring Boot, Angular and TypeScript by Kai Tödter
- The Ultimate Asynchronous Stack by Erwin de Gier & Roy Straub

Where the Java community meets.
jaxlondon.com
The Conference for Java & Software Innovation
Oct 9 – 12, 2017 | London

**OCT. 9**

**09:00** - 17:00
**Workshop: The Ultimate Asynchronous Stack**
Erwin de Gier & Roy Straub, Sogeti

**09:00** - 17:00
**Workshop: Cool Web Apps with Spring Boot, Angular and TypeScript**
Kai Tödter, Siemens AG

**09:00** - 17:00
**Workshop: Architecture with Agility**
Kevlin Henney, independent consultant, speaker, writer and trainer

**09:00** - 17:00
**Workshop: Running Production Microservices**
Russ Miles, Atomist

**09:00** - 17:00
**NodeJS 101 Workshop**
Gil Tayar, Create School

**10:00** - 10:50
**Better software development systems through stress; the hidden ingredient behind learning and continuous improvement**
Russ Miles, Atomist

**10:50**
**COFFEE BREAK**

**11:30** - 12:20
**How to Build a Microservices Infrastructure in 7 Days**
Gil Tayar, Create School

**11:30** - 12:20
**Turbo Charge CPU Utilization in Fork/Join Using the Managed-Blocker**
Dr Heinz Kabutz, JavaSpecialists.eu

**11:30** - 12:20
**Eclipse MicroProfile: Accelerating the adoption of Java Microservices**
Emily Jiang, IBM

**11:30** - 12:20
**Apache Kafka, Enterprise Architecture, and Streaming Data: What we can learn from Heraclitus**
Tim Berglund, Confluent

**11:30** - 12:20
**DevOps Driven Development and Delivery**
Helen Beal, Ranger4

**12:20**
**LUNCH BREAK**

**14:30** - 15:00
**Blockchain: Why Open Source and Collaboration are Crucial for Tech**
Brian Behlendorf, Hyperledger Project

**15:15** - 16:05
**All Things Data**
Daniel Lebrero, IG

**15:15** - 16:05
**RESTful Hypermedia APIs**
Kai Tödter, Siemens AG

**15:15** - 16:05
**Continuous Delivery with Containers: The Good, the Bad, and the Ugly**
Daniel Bryant, Big Picture Tech Ltd

**15:15** - 16:05
**Java 9: Escaping JHell with JShell**
Dr Raoul-Gabriel Urma & Dr Richard Warburton, Iteratr Learning

**15:15** - 16:05
**Real Kafka Implementations of the Enterprise**
Tim Berglund, Confluent

**16:05**
**COFFEE BREAK**

**16:45**
**Load Testing Like a Pro**
Rob Harrop, Skipjaq

**16:45**
**Resilient Microservices with Kubernetes**
Mete Atamel, Google

**16:45** - 17:35
**Java 9: Beyond Contention!**
Monica Beckwith, Code Karam LLC

**16:45** - 17:35
**Finally, Enterprise JavaScript Is Easy!**
Geertjan Wielenga, Oracle

**17:35**
**COFFEE BREAK**

**18:15**
**A Practical Guide to Docker and µService Deployments**
Davide Fiorentino Io Regio, United Nations et al.

**18:15** - 19:05
**Architecture with Agility**
Kevlin Henney, independent consultant, speaker, writer and trainer

**OCT. 10**

**08:00**
**CONFERENCE CHECK-IN**

**08:00** - 09:45
**The Error of Our Ways**
Kevlin Henney, independent consultant, speaker, writer and trainer

**10:00** - 10:50
**The Ultimate Java Microframework Shoutout**
Erwin de Gier, Sogeti

**10:00** - 10:50
**The Road to Continuous Deployment: A Case Study**
Michiel Rook, make.io

**10:00** - 10:50
**Lambdas: It’s Java Jim, but not as we know it**
Simon Ritter, Azul Systems

**10:00** - 10:50
**Deep Learning Anomalies with TensorFlow and Apache Spark**
Khanderao Kand, Oracle

**10:00** - 10:50
**COFFEE BREAK**

**11:30** - 12:20
**Apache Kafka, Enterprise Architecture, and Streaming Data: What we can learn from Heraclitus**
Tim Berglund, Confluent

**11:30** - 12:20
**DevOps Driven Development and Delivery**
Helen Beal, Ranger4

**12:20**
**LUNCH BREAK**

**14:30** - 15:00
**Blockchain: Why Open Source and Collaboration are Crucial for Tech**
Brian Behlendorf, Hyperledger Project

**15:15** - 16:05
**All Things Data**
Daniel Lebrero, IG

**15:15** - 16:05
**RESTful Hypermedia APIs**
Kai Tödter, Siemens AG

**15:15** - 16:05
**Continuous Delivery with Containers: The Good, the Bad, and the Ugly**
Daniel Bryant, Big Picture Tech Ltd

**15:15** - 16:05
**Java 9: Escaping JHell with JShell**
Dr Raoul-Gabriel Urma & Dr Richard Warburton, Iteratr Learning

**15:15** - 16:05
**Real Kafka Implementations of the Enterprise**
Tim Berglund, Confluent

**16:05**
**COFFEE BREAK**

**16:45**
**Load Testing Like a Pro**
Rob Harrop, Skipjaq

**16:45**
**Resilient Microservices with Kubernetes**
Mete Atamel, Google

**16:45** - 17:35
**Java 9: Beyond Contention!**
Monica Beckwith, Code Karam LLC

**16:45** - 17:35
**Finally, Enterprise JavaScript Is Easy!**
Geertjan Wielenga, Oracle

**17:35**
**COFFEE BREAK**

**18:15**
**A Practical Guide to Docker and µService Deployments**
Davide Fiorentino Io Regio, United Nations et al.

**18:15** - 19:05
**Architecture with Agility**
Kevlin Henney, independent consultant, speaker, writer and trainer
Agile Machine Learning: From Theory to Production
Rob Hinds, Basement Crowd; Sumanas Sarma, Basement Crowd/Queen Mary University of London

Simple and Modern Concurrency in Java: Completing the Promise
Dr Raoul-Gabriel Urma & Dr Richard Warburton, Iteratr Learning

Fly High with Microservices Architecture
Angel Gruev, Dreamix Ltd

OCT. 11
08:00 CONFERENCE CHECK-IN
09:00 The Long Road
Sandro Mancuso, Codurance

09:45 COFFEE BREAK

10:30 Measuring DevOps: the Key Metrics that Matter
Anders Wallgren, Electric Cloud

10:30 REPL Driven Development: Immediate Feedback for your Serverside Code
Daniel Lebrero, IG

10:30 Kotlin and Twitter: How to work with streaming data
Roberto Franchini, OrientDB LTD

10:30 Complicating Complexity: Algorithm Performance in the New Machine Age
Maurice Naftalin, Author, Programmer, Teacher

11:20 COFFEE BREAK

12:00 CQRS & Event Sourcing in the Wild
Michiel Rook, make.io

12:00 A Breathless Tour of Blockchain
Eoin Woods, Endava

12:00 Keep the JVM – Ditch Java
Dr Russel Winder

12:00 Java SE 9 Modules: an Introduction
Stephen Colebourne, OpenGamma

12:00 G1GC: The Dark Horse
Monica Beckwith, Code Karam LLC

12:50 LUNCH BREAK

15:00 Porting Doom 3 to Java: Lessons Learned
Mahmoud Abdelghany, Blue4IT

15:00 Server-side Swift for Java Developers
Ian Partridge, IBM

15:00 Using Ethereum to build Java applications on the blockchain
Conor Svensson, BLK technologies

15:00 The monster coming over the hill – enterprise coding is dead
Sarah Saunders, capgemini

15:00 Busy Java Developer’s Guide to TypeScript
Ted Neward, Neward & Associates

15:50 COFFEE BREAK

16:30 Lessons to Build your Software Career: The Secret of Miyagi
Armagan Amcalar, unu GmbH

16:30 Scaling Event Sourcing for the IoT and Mobile
Lorenzo Nicora, OpenCredo

16:30 Automated Data-driven Deployments
Malcolm Sparks, JUXT

OCT. 12
09:00 Workshop: Developing Java applications on the Blockchain with web3j
Conor Svensson, BLK.io

09:00 Busy Java Developer’s Workshop on DDD with Apache Isis
Ted Neward, Neward & Associates
We talked with Marcus Biel, one of JAX London’s Top 20 Java Influencers, about the current state of Java and the future of modularity. In this interview, Biel explains why Java isn’t going anywhere, weighs in on Java 9 and the modular ecosystem and reveals what he would like to see in Java 10.

JAX Magazine: Some people seem to believe that Node.js might have a chance at overtaking Java in the near future. Can Java really be dethroned?

Marcus Biel: To me, this is a comparison that really doesn’t matter, a comparison of apple and pears. Java is best suited for complex (multithreaded) business systems; Node.js is great for real-time web applications. Ask a Java developer and he/she will favor Java; ask a Node.js developer, and he/she will favor Node.js. There is room for both.

JAX Magazine: This year, Stanford’s famous introductory course for programming dropped Java in favor of JavaScript. What does this say about the relevance and popularity of Java?

Biel: To me, the first question is what does this say about Stanford? Stanford believes that learning Java is bad as a first programming language. That’s all this says, and it’s okay if they think that. For me, Java was my first programming language at university. As a Clean Code enthusiast, from the first minute, I saw it as a great advantage that Java is statically typed.

JAX Magazine: What did you think of the fact that Java 9 was delayed to September? Do you agree with the JCP Executive Committee’s decision to vote against JSR 376?

Biel: Well, that is very difficult to say. Jigsaw is a big chance, it will hurt, but we will benefit from it in the long run. It might be worth the wait to make it as painless as possible.

JAX Magazine: Georges Saab, chairperson of the OpenJDK governing board and vice president of development for the Java Platform group at Oracle told us in early June that many developers will probably get started on JDK 9 without modules. How do you feel about the modular ecosystem?

Biel: Jigsaw is a big chance. It will require a lot of work to migrate, and the majority of companies will delay doing that. The majority of companies will, for many years, see this as a great risk and delay it for as long as (they think) possible –
without considering the great advantages that modularization entails. It’s very unfortunate.

On top of this, if you consider that most companies are still using Java 7 and only a few companies are actually using Java 8, I think that this topic will still be relevant for the next 5–10 years.

**JAXmag: What is the most important misconception about Java 9? How about Jigsaw?**

Biel: As with all changes, the misconception is that upgrading to Java 9 or migrating to Jigsaw is more expensive and riskier than not doing it. Every upgrade involves a risk, it’s true, but sooner or later you will have to upgrade anyway. The sooner you upgrade, the better, because the sooner you can benefit from all its advantages. In the case of Jigsaw, you’ll benefit from the advantage of proper encapsulation on module level. Jigsaw will allow us to build modular systems, with a much clearer, more precise structure, which will be understood much faster by new developers joining the team or by business people.

**JAXmag: What would you like to see in Java 10?**

Biel: Well, if I was granted a wish, removing features and revising wrong decisions of the past is much more important to me than adding new features. I would love to see some new changes that would not be backward compatible. For instance, I would love to see Checked Exceptions removed, but this is not going to happen too soon. Anyway, to list a few new features that would be nice to see with Java 10:

- Pattern Patching
- Value Objects
- Multiline Strings
- List<int>

To read the complete interview series with 11 Java influencers, check out [www.JAXenter.com](http://www.JAXenter.com).
Eclipse Oxygen is here but even though this milestone has been successfully reached, the development continues. We talked to Mike Milinkovich, Executive Director of the Eclipse Foundation about his favorite Eclipse Oxygen features, the upcoming Java 9 support and the future of the classic IDEs.

JAX Magazine: You have been at the helm of the Eclipse Foundation since its inception in 2004. The first Eclipse release train started in 2006 with Callisto, where 10 projects adjusted their release schedules (or even earlier in 2004 resp. Eclipse 3.0). In 2017, we have 83 projects joining the Oxygen release train. If you look back and compare the two releases – what was your main challenge or focus point in 2006 and how are things in 2017?

Mike Milinkovich: In 2006, a lot of our challenges were simply related to IT and IP resources. Getting all of the intellectual property reviews and analyses done was a challenge for the team. And dealing with the tsunami of download requests when Callisto shipped was a major challenge for our IT staff, and for the limited bandwidth that we could afford at that time.

In 2017, the simultaneous release is a well-oiled machine from the perspective of the Eclipse Foundation. The biggest challenge that we experienced was having staff and community resources stepping up to replace the role that David Williams played for many years as the Chair of the Planning Council, and as an extraordinary release manager.

JAXmag: Do you still find time to contribute to projects that are part of the Eclipse ecosystem? And if not, which ones would you really like to participate in?

Milinkovich: I don’t contribute to any Eclipse projects, but I do use a number of the Eclipse IoT technologies for hobby projects around the house. I guess if I were to participate in any of them, it would be the ones that I could play with on one of my Raspberry Pis, like Eclipse Kura or Eclipse Smart-home.

JAXmag: What are your personal technological highlights of Eclipse Oxygen?

Milinkovich: For me personally, it is the one which didn’t actually ship as part of Oxygen: Java 9 support. I know how hard the Eclipse Java Development Tools (JDT) team has worked on building great Java 9 support, and I encourage users to try out the beta support for it [1].

JAXmag: Before Eclipse, there were a lot of different editors, IDEs and development tools – most of them were incompatible with each other. The Eclipse platform consolidated this market to a point where most of these old tools vanished and only three IDEs for Java persisted. Nowadays, we see new IDEs for Java coming out, like Eclipse Orion, Eclipse Che, VS Code, Theia, OpenShift.io etc. Do you think this trend towards more diversity leads us back in the old times of tool silos?

Milinkovich: I actually think of this phenomenon differently. In my view, there are two parallel trends occurring:

Personal highlight of Eclipse Oxygen: Java 9 support

Portrait

Mike Milinkovich is the Executive Director of the Eclipse Foundation. Outside of work, Mike’s passions are his family, the family cottage and hockey (as a coach, player and fan) in pretty much that order. When he’s not working, or traveling for work, you will probably find him involved in one of those three things.

https://mmilinkov.wordpress.com
@mmilinkov
1. For quite a few years, Java was the main language and platform used in software development. Its dominance from (say) 1998 to 2008 would be hard to overstate. Now we have a very different, and much more polyglot and fractured world where developers need to deal with numerous languages, frameworks, runtimes, and platforms in order to ship a complete system. The amount of effort to configure and maintain a complex IDE that covers all of a full-stack developer’s requirements is far too large. As a result, developers have been forced in many cases to leave IDEs entirely, adopting simple language-aware editors. I think of Atom and VS Code as falling into this category.

2. The second trend is the migration of developer tools to the cloud. Already, most of the tools that developers use are on the web, and their IDE is the last major component to move. I believe that this is going to happen over the next couple of years. The interesting thing to note here is how cluttered the market is. The current state of cloud-based developer tools reminds me very much of what the desktop IDE world looked like before the Eclipse IDE platform came along, and I believe that the market is ripe for consolidation around an open source platform like Eclipse Che.

**JAXmag:** The technology landscape has changed a lot in the last decade. Now, we see a movement towards container technologies, the cloud, serverless, mobile, IoT etc. How does Eclipse Oxygen respond to these new trends and do you think the classic IDE will still be of interest in the future?

**Milinkovich:** There will be desktop IDEs for a very very long time. For the right applications, they cannot be beaten for developer productivity. In particular, I think that in multi-platform spaces such as Mobile and IoT, we will observe the continued adoption and success of desktop IDEs.

However, I do believe that cloud, serverless, and container technologies are going to drive developers to look for cloud-based IDEs which integrate the developer experience into their DevOps environment. Again, Eclipse Che with its ability to use containers for its workspaces, and its integration with Kubernetes really shines in this space. Red Hat’s use of Eclipse Che to create its OpenShift.io offering is a great example that I expect to see repeated in the future.

**JAXmag:** Do you think that there will be a complete overhaul of the underlying technology of the Eclipse IDE at some point of time?

**Milinkovich:** That is entirely up to the community and project teams to decide.

**Java 9 support in Eclipse Oxygen**

**JAXmag:** The Eclipse Planning Council had discussed possibly moving the release of Eclipse Oxygen to align with the anticipated Java 9 release date [2]. Are you glad that you dismissed this thought, now that Java 9 has been delayed?

**Milinkovich:** Absolutely. It was awfully nice of Oracle to line up their Java 9 release date with Oxygen.1.

**JAXmag:** Will there be an Eclipse update in September [3]?

**Milinkovich:** Yup! But to be clear, there was always going to be an update to Eclipse Oxygen in September. It was simply some good luck that the dates ultimately aligned.

**JAXmag:** The technology landscape is not the only thing that changed – the role of the Eclipse Foundation has also changed. What was your understanding of your role back in 2004, what is it now?

**Milinkovich:** In 2004, the focus was entirely on the development tools landscape, and in particular, on desktop IDEs. Over the past thirteen years, the Eclipse Foundation and the Eclipse community have evolved to cover an enormous spectrum of technologies.

In one sense, my role is exactly the same: to be the cheerleader-in-chief for the Eclipse projects, community, and members. But on the other hand, the range of technologies that we are dealing with has greatly expanded. From the Internet of Things to geospatial big data, from quantum computing to systems modeling, the Eclipse community is working on world-leading open source technology.

**JAXmag:** Are there new aspects or initiatives coming next year to the Eclipse Foundation?

**Milinkovich:** Sometime in the next few months, a new version of the Eclipse Public License will be published. The EPLv2 will be a modern copyleft license that contains many enhancements over the original. So rolling out this new license, and migrating the Eclipse Foundation’s projects over to use it will be a large focus for our staff and I. Some of the new features of the EPLv2 that are important include:

- Clearer relevance for scripting languages like JavaScript.
- Allowing new projects to be compatible with the GPL if they choose to do so.
- Replacing the concept of “module” with “file” to be more consistent with common usage.
- Removing mention of New York and USA law.

References


Get your Java errors under control with error monitoring

Java is not without its potential pitfalls

Java became a go-to language for Web-facing applications and Internet projects. However, the use of Java is not without its potential pitfalls, and that’s something developers should keep in mind. In this article, AJ Philips teaches you how to get your Java errors under control with error monitoring.

by AJ Philips

The Java programming language is immensely popular with developers and has been for many years. It’s not just a case of having another object-oriented programming language to build applications with – Java’s rise coincided with the emergence of embedded web programming. Java became a go-to language for Web-facing applications and Internet projects. However, the use of Java is not without its potential pitfalls, and that’s something that’s important for developers to keep in mind.

The root of the problem(s)

Some of the common code problems in Java relate to the semantics that programmers use to build a code base. Without the right syntax, compiler errors and other errors can result.

Consider the omission of a closing bracket or parenthesis on a declaration or command. This type of mistake will generate an “expected” error and has to be fixed for proper execution. Another similar error is the “unclosed string error” where a string is not closed out with a quotation mark.

Another common error called “incompatible types” happens when integers or strings or other data types are used.
improperly, or combined in ways that don’t work for the program. Trying to assign one type of data type variable to another may create an “incompatible type” error. Other malformed syntax can produce an error called “invalid method declaration” or an “unreachable statement” error, or one stating an operator cannot be applied correctly.

What all of these mistakes have in common is that they arise from syntax that’s not properly and precisely controlled. It only takes one keystroke to make the errors, and they’re a minefield for programmers who may be meticulous in their writing, but still experience the occasional misplaced character while typing.

Automating the error correction

In the early days of object-oriented programming, there weren’t a lot of tools to catch the errors. Code base work could be tedious and labor-intensive – programmers had to catch the errors or fight them when trying to compile the program. This generated a lot of protocol in programming offices and oversight of individual programmers and their work.

Today’s developers have other options – the advance of digital analytics means programs can be created to automate some of the error correction that used to be done by hand.

These types of automation programs are immensely valuable in developer communities. Developers understand that they can bring products to market more quickly, fine-tune a codebase, and work more efficiently with tools that feature automated processes. Many of these tools also have combined features offering more research capability, so that developers can work better on the fly and do various kinds of required investigation while they are putting code together.

Improving the world of Java

The Stackify platform [1] has a lot of this valuable functionality in place. Stackify looks at blogs, monitoring, metrics and available tools, and offers developers real assistance or getting where they need to go. The company calls the products “a magical developer map” in which professionals can find problems quickly and solve them actively and decisively.

When it comes to assisting developers, insight and transparency are key. Troubleshooting application problems can lead developers down some very dark paths – and without modern tools like Stackify, troubleshooting can take a lot of time. However, with these new tools and platforms, there is a way through these complicated processes. Stackify Retrace helps developers to effectively retrace what the code is doing so that bugs and glitches have nowhere to hide.

References

[1] https://stackify.com/

AJ Philips lead many automation projects and has consistently managed to provide value through removing manual work out of different processes. In his spare time, he likes to cover tech and automation related topics and share his know-how.
Abby Kearns, Executive Director of Cloud Foundry Foundation

“Diversity ensures continuous innovation”

Women are underrepresented in the tech sector – myth or reality? In addition to the Women in Tech survey, we also launched a diversity series aimed at bringing the most inspirational and powerful women in the tech scene to your attention. Today, we’d like you to meet Abby Kearns, Executive Director of Cloud Foundry Foundation.

Abby Kearns, Executive Director at Cloud Foundry Foundation

Abby Kearns is a true tech veteran, with an 18-year career spanning product marketing, product management and consulting at a mix of Fortune 500 and startup companies.

As the first fellow at Cloud Foundry Foundation and VP of Strategy, Abby was responsible for structuring and executing operational and strategic initiatives, as well as leading the User Advisory Board and Industry Special Interest Groups. Prior to joining the Foundation, she was part of the Product Management team at Pivotal, focusing on Pivotal Cloud Foundry. Previously, Abby led a Product Management and Product Marketing team at Verizon focused on cloud services.

Abby’s love for technology was there all along

I have taken a varied career path. After I graduated college, I wanted a job in technology and ended up working at Sabre as a project manager. I led a project implementing a new email system for an airline. It was Novell GroupWise – if I’m going to date myself. I quickly learned a lot about infrastructure, project and program management, as well as how to lead a team that was large and distributed.

My learnings went on to shape my career. From there, I spent the next several years leading teams, architecting and implementing infrastructure projects, as well as operations.

Operations teach you a lot about what “Day 2” of a technology solution really means and provide perspective on what happens after something goes “live”. You end up having enormous empathy for the user. That role really helped shape how I think about any project, even now. I ask questions like: What happens once this is in production? What is the user experience? Is this something that is going to scale?

I have always loved technology and found comfort not only in what it can do but in its potential to change and evolve. I am someone who enjoys learning and will continue to learn throughout the rest of my life. Technology is the perfect outlet for this because I constantly need to learn what is new and different to stay current.

Persistence is key

I was the first person in my family to go to college, and the first (and only) person to have a career in technology. So, I did not really have a role model growing up for this type of career. I just knew this is what I wanted to do, and I persisted. I worked several jobs throughout college to pay for school and then taught myself how a computer worked, and how to put one together. I am really goal-oriented and stubborn, so when I set my mind to something I focus on accomplishing it.

Abby is enthusiastic for the next twenty years in technology because

For a woman in technology, there is a tremendous amount of obstacles – some of which I have only recently realized. When I first started in my career, there was a lot of acceptance of bad behavior because “it was the way the world worked”. As a woman, you had to work harder and be better than your male counterpart just to reach equal footing.

The world is finally starting to change. We are finally starting to believe the world does not actually “work” that way. I am really enthusiastic for the next twenty years in technology. A diverse group of people can bring so much power to our industry.
Today, I am privileged to have the role of Executive Director of the Cloud Foundry Foundation, the open source software foundation charged with holding the intellectual property for Cloud Foundry, an open source Cloud Application Platform. As a foundation, we are responsible for ensuring the continued success of the open source project, as well as ensuring that it has a sustainable ecosystem and community around it.

This job offers an amazing opportunity to work with a brilliant team, an exceptional board of directors and a community that continues to inspire me with its talent and drive. I’ve worked with incredible people and technology throughout the years. It hasn’t always been easy but perseverance in the face of any challenge has always been my strength. In my current role, I make an impact with our member companies, have the opportunity to learn from an incredible board, solve hard problems, and work with an exceptional team. But more importantly, I am having fun!

Being a woman in technology
Many women don’t receive support, whether they have just made the decision to major in a technical area or have been in technology for years. When you head down any career path, it can be challenging to find mentors who are the right fit and who are open to taking on mentees to support.

Some organizations are old boys’ clubs – with few if any women in leadership positions and it can be incredibly hard to make a cultural shift unless it is from the top down.

There is a lack of inclusion to which men can be blind. I recently heard Kara Swisher speak about interviewing male CEOs who had never experienced this challenge, so they didn’t believe how deep of a problem it is. Of course, there are systemic issues within our wider society, such as a lack of consideration for working mothers or insufficient maternity leave policies. The more challenges women face, the weaker the pipeline becomes. If women lack support, there will be fewer women looking to enter the industry, period. We must continue to improve this.

Women in STEM
It’s a vicious cycle: if more women worked in STEM, they wouldn’t face this same level of exclusion. If they didn’t face exclusion, more women would work in STEM. More diverse perspectives – not only women’s but those from all types of unique backgrounds – offer up more voices and opinions, and provide greater opportunities to solve problems collaboratively.

Where is the diversity discussion headed?
This is not an easy problem to solve. Fortunately, there has been traction in recent years with organizations that have made it a priority to bring in executives who identify with traditionally underrepresented groups. Diversity is key to moving the industry forward; it helps us foster a sustainable and open community, and ensures continuous innovation.

We focus on inclusion here at the Cloud Foundry Foundation, such as bringing the best and brightest speakers to our events so that a diverse set of perspectives are shared. This year, we actually have more women giving keynotes than men at our upcoming Silicon Valley Summit, which sadly, is not typical in this industry.

Tips & tricks
• Be confident in your own abilities.
• Be vocal and speak up for yourself. You bring as much to the table as your male counterparts.
• Be confident in your own abilities.
• Know your value and trust your gut.

That is advice I would give to any woman in the workforce, whether or not she works in technology.

Don’t miss our Women in Tech profiles. You’ll find them all online at www.JAXenter.com.

Abby Kears is a true tech veteran, with an 18 year career spanning product marketing, product management and consulting at a mix of Fortune 500 and startup companies. As the first fellow at Cloud Foundry Foundation and VP of Strategy, Abby was responsible for structuring and executing operational and strategic initiatives, as well as leading the User Advisory Board and Industry Special Interest Groups. Prior to joining the Foundation, she was part of the Product Management team at Pivotal, focusing on Pivotal Cloud Foundry. Previously, Abby led a Product Management and Product Marketing team at Verizon focused on cloud services. In her free time, Abby enjoys posting up at her local coffee shop, indulging in food and wine, and spending time with her husband and son.

Imprint
Publisher: Software & Support Media GmbH
Editorial Office Address: Schwedlerstraße 8, 60314 Frankfurt, Germany
www.JAXenter.com

Editor in Chief: Sebastian Meyen
Editors: Gabriela Moroc, Hartmut Schlosser
Authors: Monica Beckwith, Brian Behlendorf, Marcus Biel, Daniel Burant, Stephen Colebourne, Kevlin Henney, Abby Kears, Mike Milinkovich, AJ Phillips, Georges Saab, Gil Tavay, Kai Tödter, Anders Wallgren, Eoin Woods
Copy Editor: Nicole Bechtel, Frauke Pesch
Creative Director: Jens Mainz
Layout: Flora Feher

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

Entire contents copyright © 2017 Software & Support Media GmbH. All rights reserved. No part of this publication may be reproduced, redistributed, posted online, or reused by any means in any form, including print, electronic, photocopy, internal network, Web or any other method, without prior written permission of Software & Support Media GmbH.

The views expressed are solely those of the authors and do not reflect the views or position of their firm, any of their clients, or Publisher. Regarding the information, Publisher disclaims all warranties as to the accuracy, completeness, or adequacy of any information, and is not responsible for any errors, omissions, inadequacies, misuse, or the consequences of using any information provided by Publisher. Rights of disposal of rewarded articles belong to Publisher. All mentioned trademarks and service marks are copyrighted by their respective owners.

www.JAXenter.com | August 2017 37