FACILITATING THE SPREAD OF KNOWLEDGE AND INNOVATION IN PROFESSIONAL SOFTWARE DEVELOPMENT

PERSPECTIVES ON GDPR

eMag Issue 60 - Apr 2018

IN THIS ISSUE

FOLLOW US


facebook.com /InfoQ


@InfoQ


google.com /+InfoQ


linkedin.com company/infoq

CONTACT US

GENERAL FEEDBACK feedback@infoq.com
ADVERTISING sales@infoq.com
EDITORIAL editors@infoq.com

A LETTER FROM THE EDITOR

Manuel Pais

Until the EU's General Data Protection Regulation (GDPR) not only comes into effect, but actually produces a body of precedents on what exactly constitutes a violation subject to a fine (which, remember, can be as high as €20 million or 4% of annual global turnover, whichever is greater), the margin for interpreting which procedures and oversight needs to be in place will remain wide. What we do know for now is that companies will have to adopt an explicit approach to data governance, including data profiling, data quality, data lineage, data masking, test-data management, data analysis, and data archives.

As data scientists, software engineers, and operators, we need to actively cover all bases, not only because the GDPR tell us to, but also because it makes business sense to establish proactive data governance and adequate privacy controls. Our customers will be thankful, our reputation will be safeguarded, and crippling fines will be avoided. This e-mag specifically addresses those three overlapping but distinct perspectives on the impact of the GDPR.

As Jon Topper candidly told me while discussing the outline of his article, "people should have been doing a bunch of this stuff already". Many organizations have noticed the increasing frequency of data-breach scandals and harm recently caused. Many others are being driven by the GDPR to rethink their practices (or lack thereof). This is a good thing.

As Andrew Burt, chief privacy officer and legal engineer at Immuta, highlights in his article, we can expect to see some organizations looking to simply tick all the boxes for GDPR compliance, while more data-driven organizations will be looking at the GDPR as an opportunity to rethink their whole data strategy. Regardless, they all need to focus on what customer data they hold across all data storage, when that data was collected, who has access to it (read or write), and what purpose it serves. They must be able to remove any piece of data upon customer request, in an auditable fashion.

Arto Santala, a software architect at Solita Oy, warns software engineers of the need to "build privacy" in their applications and apply a principle of "least access" when dealing with customer data to avoid being classified as data processors. Santala stresses the need to not go overboard but to apply a risk-based approach to GDPR compliance, based on number of customers, business impact of a data breach, and the amount of sensitive data in the system. The risk level will help determine how comprehensive you need to be in practices like keeping audit trails that support non-repudiation (cannot be altered even by administrators), protecting data both at rest (in databases) and in traffic (over the network), anonymizing test data, and making sure that logs do not contain personally identifiable information.

Jon Topper, CTO at The Scale Factory, provides clear hands-on advice on how to operationalize GDPR requirements: build strong identity concepts into your infrastructure (machines, containers, applications); use role-based access control and apply the principle of "least privilege"; clarify your backup retention policies so you can inform customers and auditors how long a piece of data will remain in a backup; don't copy production data into dev/test environments without anonymizing personally identifiable data; and keep abreast of modern security and monitoring tools that allow faster response to incidents or suspicious patterns.

If you are relatively new to the subject, I recommend starting with the Q&A with Immuta on the implications of the GDPR. This provides a good overview of where the GDPR is coming from, what it is about, and how to approach compliance and data governance.

CONTRIBUTORS

Manuel Pais

is a DevOps and Delivery Consultant, focused on teams and flow. Manuel helps organizations adopt test automation and continuous delivery, as well as understand DevOps from both technical and human perspectives. Co-curator of DevOpsTopologies.com. DevOps lead editor for InfoQ. Co-founder of DevOps Lisbon meetup. Co-author of the upcoming book "Team Guide to Software Releasability". Tweets @manupaisable

Jon Topper

is CTO at The Scale Factory, a UK-based DevOps and cloud-infrastructure consultancy. He's worked with Linux web-hosting infrastructure since 1999, originally for ISP Designer Servers, and then for mobile-technology startup Trutap. Since 2009, Topper and The Scale Factory have worked to design, build, scale, and operate infrastructure for a range of workload types. Topper has a particular interest in systems architecture and in building high-performing teams. He's a regular speaker on DevOps and cloud infrastructure, in the UK and internationally.

Arto Santala

is a software architect at Solita Oy. He has more than 20 years of experience in crafting software solutions to enable tomorrow›s world right now. His greatest passions include automating just about everything, and using agile methodologies to get the right things done just right.

Andrew Burt

is chief privacy officer and legal engineer at Immuta, where his focus is on automating regulatory compliance within big-data environments. He is also a visiting fellow at Yale Law School's Information Society Project. Previously, Andrew served as special advisor for policy to the head of the FBI Cyber Division, where he served as chief compliance and privacy officer for the division, and as lead author on the FBI's after-action report for the 2014 attack on Sony.

Steve Touw

is co-founder and CTO of Immuta, a unified data platform for the world›s most secure organizations. Touw has a long history of designing large-scale geo-temporal analytics across the US intelligence community, to include some of the very first Hadoop analytics and frameworks to manage complex multi-tenant data policy controls. This experience drove him and his co-founders at Immuta to build a software product that frees data-science teams to access and work with high-value data.

Read online on InfoQ

KEY TAKEAWAYS

The EU's GDPR is the most forward-leading privacy regime on the planet, with fines of up to 4% of global revenue.

According to a study, a whopping 96% of companies admit to not understanding the GDPR.

The GDPR is about data privacy (when, what, and why data is accessible), and that means the classic encryption and role-based access controls that organizations have been relying on won't cut it.

There must be a data abstraction layer to enforce the audit and privacy of the data.

Organizations need to divide their data governance model into three buckets: collection (focused on user accounting), usage (focused on managing access controls), and audit (focused on GDPR traceability requirements)

Q&A with Immuta on the Implications of the GDPR

by Manuel Pais

The European Union approved a new General Data Protection Regulation (GDPR) in late April 2016, which will come into effect in May 2018. This regulation will require companies to adopt a comprehensive data governance approach, including data profiling, data quality, data lineage, data masking, test-data management, data analysis, and data archives.

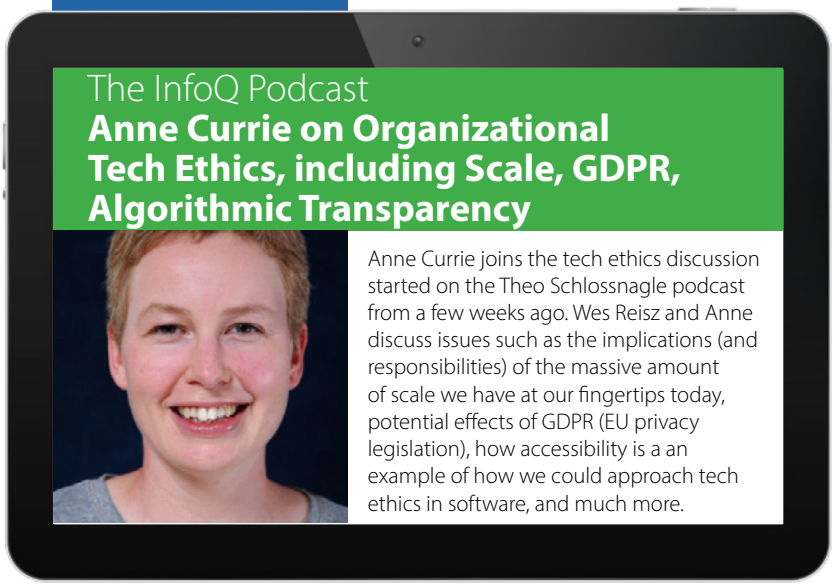
The reach of data protection will extend to genetic data, e-mail addresses, and IP addresses, to name a few. Also, users will have finer-grained rights on the protected data kept by companies. Explicit consent will be required for an organization to use individual pieces of information such as e-mail address or phone number, and their combined use will also require explicit consent.

A year ago, InfoQ talked with Immuta's Andrew Burt, chief privacy officer and legal engineer, and Steve Touw, chief technology officer, to better understand the implications and challenges of the GDPR.

InfoQ: Can you try to summarize what GDPR is and why organizations should care about it?

Andrew Burt: The General Data Protection Regulation is the EU's primary data-governance regulation and applies to any business using data from EU data subjects. It is the most forward-leading privacy regime on the planet, with fines of up to 4% of global revenue. With such staggering fines, breaching the GDPR is a risk that many enterprises quite literally may not be able to afford.

InfoQ recommends



InfoQ: So we're less than a year and a half away from this legislation coming into effect in the EU. How prepared are organizations to address it?

Burt: Not very. There have been some studies on this recently, and surveys of those charged with data privacy and security in large organizations illustrate that they're just waking up to the risk they face. According to one such study conducted by Dell, about 70% of the professionals they surveyed reported that their organization was either not ready or unaware of how to prepare for GDPR. Only 3% reported having any type of plan at all. According to another study by Symantec, a whopping 96% of companies admit to not understanding the GDPR. And that says a lot. How can organizations prepare for the GDPR when they don't understand it? 2017 is really shaping up to be the year that enterprises start to realize the compliance risk they face and start looking for comprehensive solutions.

InfoQ: What is the number-one cost or penalty for organizations failing to comply with regulations?

Burt: The number-one cost is the potential fine of 4% of global revenue. To put that into perspective, global revenue for Apple was over $200 billion each of the last two years. So privacy violations under the GDPR could cost Apple upwards of $8 billion. By contrast, privacy violations in the US frequently incur fines that range from hundreds of thousands to a few million dollars. The GDPR is more serious than other data regulations by orders of magnitude.

Aside from monetary fines, though, it's also important to put the GDPR into perspective and ask: what is the regulation protecting? The answer for many businesses is trust — between them and their customers, between them and other business, and really throughout the data landscape. The cost of a breach isn't just financial; it can cause serious reputational damage and lead to a breakdown in trust between the enterprise and the consumer.

InfoQ: What are the biggest challenges for compliance from a technical perspective? Will it require new technologies and processes to be developed? And how will those be retrofitted or integrated with existing solutions?

Steve Touw: Some misconstrue the GDPR to be all about data security and data breaches. While data security is a big part of the regulation, it's actually a very small part of the GDPR. The GDPR is really about data privacy, and that means the classic encryption and role-based access controls that organizations have been relying on won't cut it when it comes to protecting privacy. Instead, organizations need to measure analytical "bang" against compliance "buck," and this requires measuring the analytical purpose of the data: how it's been used, for what purposes, and towards what gain. This is not as simple as aligning database roles to all possible purposes — that would be an implementation nightmare across all organizational data silos.

Instead, there must be a data abstraction layer to enforce the audit and privacy of the data, and

that's what Immuta has focused a lot of our energy and time on. Imagine, for example, an analyst in the morning working on one task, and seeing personally identifiable information because it's relevant to the purpose of that task; then, in the afternoon, the analyst switches tasks and is now seeing an anonymized version of the same data, through the same connection, for a different task because to see that level of detail isn't required or authorized. That's the type of regime we need to have in place for analysts across the enterprise.

InfoQ: In the past, we've seen strong commotion when the EU forbad the transfer of citizens' personal data outside the EU. How do you compare that to the scale of changes required by the GDPR?

Burt: The GDPR builds off of the Data Protection Directive (DPD), which is the current regulation that applies to EU data. In general, the GDPR is very similar to the DPD in its approach to privacy, but it has much sharper regulatory teeth. Regarding data transfers outside the EU, this may be one of the few areas in the GDPR where it makes things easier for many enterprises. The GDPR spells out a number of additional conditions that allow for data transfer under the GDPR that didn't exist under the DPD. Under the DPD, something referred to as "standard contractual clauses", which are EU-approved contractual clauses used to establish a certain level of safeguards between data-sharing parties, required the approval of state data-protection authorities; under the GDPR, however, no such approval is necessary. And there are a host of other modifications when it comes to transferring data between parties.

InfoQ: So we're not just talking about changes in data storage, but also data governance and even the actual roles and structure inside organizations?

Touw: Yes, while data storage is a component, it's more about the decisions on when, what, and why that data is accessible from the storage system. Typically, data is collected for an original purpose. Later, however, the data "exhaust" created from that original purpose ends up being valuable for other revenue streams. Under regulations like the GDPR, however, you may not be allowed to use that data for other purposes. And so it's important to inject users into the data workflow who understand these different regulations and can embed them within the workflow. And the GDPR is very specific in mandating that data-intensive organizations have what's called a "data-protection officer" who is empowered to make sure the regulation is being adhered to.

Burt: At Immuta, we've focused on developing an interface in our platform specifically for these types of governance personnel, to ensure that the right data is always used for the right purposes and is only seen by the right people in the right state. These controls can spur governance personnel to ask questions like "Have I anonymized the data sufficiently to use it for another purpose?" or "Is the purpose I intend to allow the data to be used for a legitimate purpose, as my organization defines it?" These can be gray areas that require humans, in some cases, to make these decisions and can also require technology to then enforce and easily and rapidly audit these decisions. If these people make the wrong decision, or if their decisions aren't implemented by technology, heads will roll.

InfoQ: Do you think online companies that operate in the EU but are based outside it are aware of the implications for them?

Burt: As a whole, we're seeing companies just starting to wake up to the risk burden created by the GDPR, both inside and outside Europe. Firms everywhere are really only now beginning to realize the extent of their liability, and because the Eurozone comprises the largest economy in the world, the GDPR really applies to any company that considers itself global.

InfoQ: Given that the regulation aims to provide users with finer-grained control of their data (including retrospectively) than we've ever seen before, how can organizations effectively and efficiently translate those requirements into a clear plan of action that is feasible yet flexible?

Touw: The first step is to divide your data governance model into three buckets: collection, usage, and audit.

Each bucket needs to sync with the others. Collection needs to focus on user accounting: what user is stored where, what has the user consented to, and how to capture consent. Usage is the most complex area in that it needs to focus on managing access controls in a consistent way, across data silos. Usage also needs to regulate when company personnel can access specific data, how will it be perturbed, masked, or restricted, what purposes it can be used for, and more. Finally, auditing requires capturing all requirements of the GDPR in terms of "data actions" so that you can build reports, sometimes within a relatively short period of time,

such as when a data subject demands to see how their data is being used in your organization.

InfoQ: Immuta has been trying to tackle some of the challenges of the GDPR and data protection in general. Could you tell us a bit more about that work?

Touw: Immuta is focused on enabling data science; if the data scientist isn't happy, enterprises aren't going to get the most out of their data. Our software platform focuses on giving data scientists a unified view into all the data they're allowed to see and use and makes that experience as easy and as empowering as possible. Under the hood, though, we've put a lot of effort into embedding laws and policies into the software so that data-usage policies are applied to the data on the fly, as it's queried, without the data scientist needing to stop what they're doing and focus on the rules. In order to do this, we provide a customizable interface for governance personnel, where they can set rules and policies across their organization and between user groups and regions.

Burt: Oftentimes, when people think about data-protection policies, they think it's just about who can see what data. But the GDPR and other regulations require so much more. Immuta's platform can enforce not only access rules (who can see what) but also purpose-based restrictions on data, which only allows users to use the data for specific reasons or specific periods of time, when certain conditions are met.

InfoQ: What advice would you give, from both a technical and a business perspective, on how to cope with these new regulations?

Burt: Start planning now. Behind the daunting rules in the GDPR are privacy and security requirements that companies will be able to meet as long as they take their planning seriously. This will require developing a roadmap both for how your organization is going to protect the data it uses, from both a collection and a usage side, and for what types of technical solutions you're going to implement to ensure you keep that data secure. You're going to want to use and invest in solutions that are secure, that are proven, and that can empower individuals in your organization and ensure compliance, without getting in the way. This technical-solution aspect is very much what we're focused on at Immuta by embedding laws and policies into our software platform so that data scientists can concentrate on maximizing the value of their data, rather than worrying about the rules.

Read online on InfoQ

KEY TAKEAWAYS

You cannot afford to ignore the GDPR, but you should not panic, either.

When creating software, you can easily expand documentation with GDPR-required details.

Privacy by default should be part of any software you craft.

Expanded user rights require some care and support.

You should revisit some software-building practices like logging.

A software designer should try to find ways to avoid being a data processor and still be able to do the work. In other words, don't unnecessarily access personally identifiable data, unless you are prepared to do so.

What Should Software Engineers Know About the GDPR?

Are you going to create new software solutions in 2018? The EU General Data Protection Regulation (GDPR) is soon moving out of the transition period to become enforceable.

Violating its terms might lead you to face fines up to €20 million — much more for large organizations. In addition to sanctions listed in the regulation, jail time is even possible for individuals responsible for great neglect or data breaches.

Obviously, this sounds severe. I have seen two extreme approaches to the GDPR: one, to pretend it does not apply to you and try to ignore it; and two, to declare that skies are falling and that no development can focus on personal data anymore. Both approaches

are misinformed and could lead to huge losses. The GDPR does not create an end-to-all-personal-data scenario; instead, it sets rules for transparent and secure handling of personal data and threatens those who ignore the rules with juicy penalties.

The GDPR strongly emphasises risk-based thinking: you take every step to mitigate privacy risks until the risks become something you can tolerate. I appreciate this regulation — there is enough software that has absolutely no security or privacy built in the design. This sort of software and its breaches lead users to mistrust how their personal data is being used. It's time to change that.

Key points to understand

This topic is huge so I am concentrating purely on the process of crafting new software solutions. There is a lot to be said about organisational support and legacy systems, but these are highly dependent on the starting point. The GDPR does not allow many exceptions to the rule, so big and small businesses, non-profits, and government organisations all need to know the main points.

One key point of the new regulation is transparency for the data subjects. When you have a registry — for example, a database — that contains personally identifiable data, the GDPR holds that its use should be transparent to the data subjects. This means that people whose data you are collecting should be able to find out what you are collecting, your purpose for collecting it, who has access to the data, and how long the data lives within the systems. To cope with this requirement, you naturally should know all these things and document them. Along with transparency, you need to provide better access to said data. Your data subjects should be able to verify, correct, export, move, and erase their data as easily as they gave it to you in the first place.

Another important topic is privacy by design/default. This should be integrated into every bit of architecture from now on. It should have been an automatic element of design before this regulation, but people often don't want to pay for security or privacy until something happens. The GDPR provides a powerful incentive to take care of this now — an incentive of up to €20 million or more. Privacy by default means a lot of things, but it essentially aims to protect personally identifiable data and its privacy, with suitable controls. This typically requires, for example, clear audit trails in the form of who did what when, including and especially read access of personally identifiable information. Additionally, you should pay attention to data when it's being stored and in transit between different layers, and apply suitable encryption to avoid data leakage from your systems.

You should also have a valid basis for processing personal data, meaning what specifically gives you the right to collect and process the information. The basis, for example, could be a law that requires you to collect and store information on individuals for a period of time. The basis for processing personal data may be a contract, agreement, or transaction.

You can ask for consent to collect and process personal data but the GDPR does not let you off easy here. It is not acceptable to have a checkbox pre-checked with a statement like "I accept that my information may be used for marketing purposes." Consent must be clear, precise, and understandable — and cannot be preset. It should be as easy to cancel consent as it is to consent in the first place. Software designers can decide none of this on their own but need to discuss it with whoever owns the software.

Here's an interesting point. If the team members that build the software have access to actual personal data while building it, they become data processors and are liable to the same sanctions and responsibilities. The same goes for the operations team. If they have access to databases and data, they are liable and responsible. You might want to think hard about that. It is possible to build and operate most systems without accessing actual customer data, after all.

Recognize personally identifiable information

The GDPR is only interested in personally identifiable information (PII). It does not apply to data that is not attached to a person, such as product or accounting information. You might still classify such data as sensitive and might still want to protect it, but the GDPR considers it non-PII data and ignores those situations.

The GDPR identifies two classes of PII data. There is data that can be used to uniquely identify a person like social-security number, e-mail address, or anything directly connected to these identifiers such as purchase history. Then there is extra-sensitive data such as medical/health information, religion, sexual orientation, or any information on or collected from a minor.

Note that combinations of information that may not be unique in isolation can potentially identify an individual, and the GDPR

PROTECT YOURSELF FROM THE DATA; PROTECT THE DATA FROM YOU.

accounts for that. So PII also includes identities that may be deduced from values like postcode, travel, or multiple locations such as places of purchase. Tiny datasets and rare combinations of values make personal identification easier.

Since any information attached to or collected from a person is protected under privacy rules, most databases are going to contain PII, with some exceptions. I would estimate 70% to 80% of typical systems data to be PII. It's not only social-security numbers and credit-card numbers that you should protect.

There's been a lot of discussion of access logs, audit logs, etc. that contain IP addresses or surrogate keys. Are these personal data? Are they registers? Do all personal rights extend to them? How strongly should they be protected? Experts seem to disagree about the answers. You have to wait and see how this evolves. I would advise, however, to avoid hysteria and to use common sense in grey areas. This sort of information could and should be protected to some extent, depending on how much harm a data breach would inflict. But I simply don't see every web server in the world becoming a PII registry in the most demanding sense of the definition.

Design for privacy

The cheapest way to have your software to comply with the GDPR is to build the requirements right in. How comprehensively you want to do this depends on the risk of the particular system in question:

- Does your system contain extra-sensitive information?
- Does your system contain something that, while not sensitive for purposes of the GDPR, would be embarrassing or dangerous to publish?
- If someone published your database content, how large a risk would that be to your business?
- How large is your database of users?

If you have few users and the information that you collect is neither sensitive nor harmful, you might consider your system a low-risk environment and use more cost-effective controls to protect it. On the other hand, if your system contains sensitive data for many users, you would want to apply stronger protection.

A good audit trail is a minimal requirement. An audit trail not only shows that you have applied controls, it also helps limit your damages in case of a data breach. After any data breach, whether by an internal or external party, the first thing you need to do is find forensics that can show which users are affected and which data was accessed. This is the information that you need to report to data-protection authorities and these are the users you may need to notify about the data breach. If you have no forensics, you may need to assume that a breach has affected all users and all records.

A good audit trail also features non-repudiation — in other words, the audit cannot be altered or damaged even by system administrators. You might want to use audit trails to see what data a system administrator was violating, for example. This has happened before, and will happen again. Audit trails are also classified as PII: they have a unique identity and data directly connected to that.

After securing robust audit trails, the next task is to limit the exposure of data. The best way to do this is to limit what data you collect and how long you store it. Introducing some kind of archival/erasure mechanisms in your software right from beginning can protect you and your users. If a data breach happens, it can only affect data that was actually in the targeted system at that point. Many systems continue to collect all data but never clean it up, even when the data becomes obsolete. The GDPR encourages you to clearly define data lifecycles and to document them. You should also restrict access to data to only what's really necessary. This is especially true for sensitive data.

I already mentioned that you should have sufficient protection mechanisms for data that's resting in a database or file system and that's moving through a network, especially to other parties. Encryption is efficient but it has its weak spots. The most powerful encryption technology encrypts early, secures your keys, and decrypts late. Unfortunately, this is a complex and costly solution to implement. Cloud services, on the other end of the spectrum, often let you cheaply and simply encrypt an entire database with a checkbox or offer to manage keys and encryption for you. While easy, these mechanisms have weak spots. You just have to find what works for you, based on risks and sensitivity of data.

It's worth mentioning that anonymisation and pseudonymisation mechanisms can help you with things like test data or analysis data. Anonymisation basically removes all identifiable information by deleting or masking fields. Pseudonymisation replaces identifiable information with pseudonyms, which typically keeps identities separate in the data. Both practices, however, are difficult to do right and may not offer perfect means to help your GDPR compatibility. Still, these are valuable tools.

You might want to revisit your logging standards and guidelines. It's easiest if you can make sure that your logs do not contain PII — otherwise, they become PII registries as well, with all the implications. Some logs are attached to individuals already: access logs and audit logs, for example. But don't pollute operational debug logs by writing user IDs, names, or similar values in them. It's good to clearly separate logs that can be linked to individuals from logs that cannot be so linked but contain general system information.

Document your systems

The GDPR loves documentation. One important point of the regulation is to be able to demonstrate compliance. You can do that by showing certificates, which in turn benefit from documenting your systems. You can build up, if necessary, from the level of documentation you are used to providing, which may vary based on many factors. But there is some additional documentation that would be useful to have from now on. Here's a brief checklist:

- Document the personal data in your system.
- Document lifecycles of collected data.
- Document all parties that process the data.
- Document your basis for collecting the data.
- Inform data subjects of their rights and explain how they can exercise them.

You should document what data you collect, your purpose, how long you store it, and your basis for processing this data. You can best do this with a combination of document types. You might (and should) already have a general policy document that explains the rules, but I've seen many software designers start to create a grid of data columns in which they can state GDPR classification. Basically, you use whatever documentation you already typically use as your domain model but then expand it with privacy information. These documents would then serve as the basis of the data-protection policy document that you would offer to your users. The first step in guaranteeing users' rights is to understand which of their information your system collects.

Another interesting facet is how the data moves over networks and which parties can access/process it. For this, you could create a data flow diagram that documents parties, tiers, and even protocols. In case of a data breach, you can use this to quickly understand and limit the exposure.

Additionally, if you wish, you might want to document what controls are used to protect the data and achieve a sufficient level of privacy.

Support expanded user rights

Most of the rights of users/data subjects already exist in the EU's established Data Protection Directive. Here's a simple list of how they look under GDPR:

- right of access,
- right of rectification,
- right to erasure,

- right to restrict/object to processing,
- right of data portability, and
- right to be notified of data breaches.

Before you start designing all kinds of crazy APIs and systems to support that, it's worth noting that the GDPR does not require these to be automated, real-time operations. In fact, you only need to respond to a request within 30 days. Responding that there is no basis to erase or export the data (because of laws, ongoing contracts, etc.) is a legitimate response — and when a person does make a request, it's very important that you identify them properly so you do not create a new data breach by manipulating or exporting some other person's data. The 30-day response window allows you to scrape or erase the data in many ways, even handling it in systems that are simply not possible to integrate.

That being said, if your organization already has a concept of digital identity for customers/users/data subjects and you provide some self-services, it's a good idea to attach these identity rights to that self-service's user interface. The more documentation you can cover with automated processes, the cheaper it becomes. Also, users are happier with real-time access, as opposed to making a request that takes 30 days to process.

It might be wise to prepare for data erasure and export functions when designing any new piece of software. You can achieve erasure by deleting information but it's easier is to partially overwrite it, effectively anonymising it. The format for data export does not seem to matter right now, but it might be a good idea to plan for it, even if your domain would not contain any GDPR user interfaces.

The most important thing to get right is the one-stop shop where data subjects can exercise their rights, leading to a process that identifies and validates the request and then to mechanisms that erase or export that data.

Data processor or not?

When you work on a software project under GDPR responsibilities, you need to answer an important question: Do you intend to be a data processor or not? By default, you would wish not to be a data processor, since being one makes you liable to any sanctions. To avoid GDPR liability, simply make sure that you will not and cannot access any personally identifiable data in any circumstances. You also need to make sure that this is clearly stated in any contracts. It might be difficult to avoid PII processing, since personal data may hide in badly written log files, test environments, and any emergency patches to your production environment. But if you wish to avoid liability, you need to resolve all this. Protect yourself from the data; protect the data from you.

Another path to take is to embrace that status of data processor. This lets you have free access to personal data, as long as you document the activity, there's valid basis for processing, and access happens within defined boundaries. This makes you clearly liable and responsible so you have to be mindful of any sanctions. But this is the route to take if you absolutely need access to PII databases.

Most software projects do not require exposure to actual PII data, and this is definitely the recommended path to take — but it might require new skills and tools.

GDPR myth-busting

No, a data subject may not erase debts or a criminal record by exercising user rights.

No, a data subject is not supposed to get everything connected to their identity when they request an export of their data. Only directly collected information is to be included. The spirit

of the regulation is transparency and the option to change service providers.

User rights are not automatically exercised. It's important to first check user identity and the validity of a request before manipulating data. This might be difficult if a database does not carry unique and secure identifiers. There might be many valid reasons to refuse a request.

No, the GDPR does not require you to encrypt everything with 2048-bit keys in rest and in transit. Controls to protect that data are only used to mitigate risks until they become accessible, and risks are different for every system and situation.

No, the GDPR does not stop you from collecting and processing user data. Take care of transparency, data security, and legal basis, and do not collect more data than you need and you should be fine.

No, having a data breach does not automatically subject you to €20 million in fines. It might — but if you have read this article and followed the advice, you should already be well on your way to lower potential penalties. Fixing what you can right now, from the start, and having a plan for the rest goes a long way. The GDPR lists about dozen questions that will be used to decide scale of penalties if that time comes.

No, sanctions are not the main reason to start doing something about data privacy. This regulation is in place because more and more data is collected every day, and more and more data breaches are happening. Having a data breach in your system can cost you much more than fees and sanctions, it can cost you your customers' trust. But the sanctions are a good way to motivate companies to spend a few euros for security and privacy when building and purchasing software and information systems.

No, cloud services are not a big no-no with the GDPR. In fact, they might actually be more in synch with privacy-by-default requirements than many traditional data centres. Of course, moving confidential data to third parties makes things mildly more complicated when it comes to contracts and documentation.

No, the GDPR does not require you to audit and log everything and have tools for intrusion detection and test-data management. Such tools might make life easier when used successfully, but the core of your approach should be risk-based assessment and suitable controls.

Conclusion

There is not a lot of time before the GDPR becomes enforceable. Already, any new systems should be built as GDPR compatible. This is not a precise definition, especially as interpretations continue to evolve and many of them will only be clarified as data breaches, audits, and sanctions occur in the future. My hope is that this article may help you to avoid being among the first to pay the price.

I think that the upcoming data-protection regulations are strongly positive and surprisingly ambitious. Finally, you have reasons to put more emphasis on security and privacy. As you improve transparency and privacy and provide more control, users of your systems will trust you more and many of them will probably happily allow you to use their information for new kinds of analysis and marketing that no one is even aware of right now. There will probably be some turmoil in summer 2018 as some of the rules will be clarified, but I believe the GDPR will lead to more security and transparency in the long run and I'm all for it.

When you find yourself in the grey areas of the regulations, unsure what to do, common sense does go a long way. Is the source of confusion something you could document and honestly explain to your software solution's users/data subjects without embarrassment or shame? If so, it's probably going to be okay. Think about worst-case scenarios like data breaches. If your database, snapshot copy, or Excel export should fall into wrong hands that publish it somewhere, how would you be able to find out exactly what was leaked, by whom, and which of your users need to be notified? Would explaining how the data was protected embarrass you? If not, you probably have done what is humanly possible, and this will probably help to mitigate the sanctions, if any. Do your best, put the rest on a roadmap. Build for a more secure world with more transparency. In the end, everybody will be happier.

Read online on InfoQ

The GDPR for Operations

KEY TAKEAWAYS

Build strong identity concepts into your infrastructure from the beginning — consider the identities of servers and containers as well as people.

Use role-based access control and the principle of least privilege to limit access to resources only to the entities that require it.

Give consideration to detailed, tamper-proof audit logs and think about your retention policies in light of the GDPR's requirements.

Stop copying data from production into development environments without first masking personal data.

Use tools from trusted vendors to monitor and respond to security events on your network.

What's the GDPR and why should I care?

The GDPR (General Data Protection Regulation) is intended to help EU citizens take greater control over their personal data. It applies to any organisation (inside or outside the EU) that holds data on EU nationals. Because its scope is so broad, this has caused some organisations an amount of panic.

Part of the reason is that the regulation introduces sanctions of up to €20 million or 4% of annual worldwide turnover. This is a more significant financial penalty than can be levied under the Data Protection Directive (which GDPR replaces), and has therefore forced more serious conversation about compliance and accountability into the boardroom.

As ever though, the real hard work of implementing the GDPR goes on elsewhere in an organisation. As practitioners of systems administration, ops, DevOps or SRE, what do you need to be worrying about?

Reading the rules

I usually recommend that if you're going to be bound by some form of regulation, you should probably take some time to read the text of the thing to grasp the idea, so that you're better positioned to discuss details with auditors or advisors. However, don't expect to find any useful technical answers in the GDPR. This is a legislative instrument designed for lawyers, not a security how-to for engineers. Readers who've worked with the prescriptive PCI-DSS standards checklists may be disappointed to find that there is no equivalent in the GDPR.

There's both good and bad news here. The good news is that many of the provisions of the GDPR are not dissimilar from the provisions of the Data Protection Directive (DPD) that it replaces. The bad news is that many organisations haven't really been doing a good job of supporting the DPD.

Fundamentally, most compliance regimes are about data security, and their content can basically be boiled down to the two main concerns of "don't let the data out" and "don't let the bad guys in". The GDPR expands on these basics, requiring that individuals be given the right to know more about how you're using their data. It allows individuals to request copies of your data about them and to request that you delete your data about them. These have obvious implications from a technology point of view.

In particular, Article 25 refers to "data protection by design and by default". What design considerations should we then be building into our systems?

Identity and access control

From a system's perspective, good security posture requires strong management of identity. Every individual interacting with a system should have a unique identity, managed centrally in some kind of directory solution. This might be a service you run yourself like Active Directory or LDAP, or it may be a SaaS solution provided by a company like Okta. This identity service will manage credentials such as passwords and any security tokens used for additional authentication factors. By providing a SAML or OAuth2 IdP (identity provider) backed by this service, you can federate this identity into any number of separate systems, allowing users to log in with a single set of credentials.

Whilst this sort of user federation is most commonly used with external SaaS services, it can also be used with internal web systems. Using Traefik or similar software (such as Google›s IAP) it's possible to put an identity-aware proxy in front of existing web apps, so that users must log into the proxy with their single-sign-on credentials in order to access the app itself.

Within Amazon Web Services (AWS), both IAM (their Identity and Access Management service) and Cognito can make use of federated sources of identity, allowing users to log in to those services with their single-sign-on credentials.

This doesn't stop with web interfaces, either: HashiCorp Vault provides a security API service that can make use of an external IdP as a source of identity for users. Once identified, users can ask Vault to vend signed SSH certificates, which will allow shell access to remote systems, or to provide temporary access credentials to databases and other services.

With a good identity story, the next thing to think about is role-based access control (RBAC). Managing permissions on a user-by-user basis is tedious, scales badly, and is difficult to reason about from an audit perspective. In an RBAC model, you create roles and assign permissions to those roles. A role might be "database administrator" or "customer-service agent", for example. Each of these roles will have very different permissions — the database administrator might have shell access to database clusters but no access to the customer-service web front end. By assigning roles to users, you can easily see who has which groups of permissions.

Identity isn't just about people — physical servers, virtual machines, containers, and applications can all have identities too. Cloud vendors such as AWS and orchestration platforms like Kubernetes all have strong concepts of identity for their component parts. In AWS, EC2 instances can gain access to their "instance identity document" as well as to a set of signatures that can be used to verify the authenticity of this data. The identity document can be used to prove instance identity to HashiCorp Vault, which can then securely provide secrets to that instance based on the role it has been assigned. A similar workflow is available to establish the identity of a Kubernetes-scheduled container. With strong identity principles like these, it should never be necessary to place secrets such as database credentials or API keys directly into application config; you can manage these centrally instead.

You can now assign roles both to people and to parts of your system. Each of these roles should be given the absolute smallest

set of permissions required for their job. This principle of least privilege makes it easy to demonstrate to a compliance auditor which people or systems may access which data objects.

Logging and auditing

Once you've established which people or systems can access which items of data, you need to record such access in order to be able to show to an auditor that your access controls are working correctly. You also need to be able to demonstrate that logs can't be tampered with after they've been written to.

In AWS, you can enable CloudTrail logging, which will log all AWS API calls made against your resources. These logs should be written and encrypted into S3 storage owned by a dedicated secure logging account. Access to this logging account should be strictly controlled; the policies on this bucket should ensure it is not possible to modify or delete logs once written.

Other system and application logs should be aggregated in a similar manner, shipped straight off host servers into secure, tamper-proof storage. The log shipper you use here should be configured to copy log data verbatim, so that you can demonstrate to an auditor that the stored logs have not been modified from their original form.

If you're also using tools such as Elastic's ELK stack to view and search log data, there are reasons why you might want to modify log data as you ship it. In that case, use a second log-shipper configuration for this less secure copy of the logs.

It's possible that logs will contain personal data as defined by GDPR terms, and as such these should be expired and deleted on an appropriate timeline. What that timeline looks like will depend on your specific workload and on any other compliance obligations you carry. Article 17 of the GDPR covers "right to erasure", by which a data subject can request that you delete any personal data you hold on them. The less data you hold by default, the easier this will be.

Article 15 covers "right of access by the data subject", wherein someone can ask you to provide all the data you hold on them. You probably have a good idea of what personal data exists in your primary data stores and how that data is linked with relations, but it might be less obvious which of those items of data could end up in your logs. This probably means that you'd need to be able to search for a particular user's log entries under a "right of access" request. In this case, structured (rather than free text) log data is likely to be useful, and search tooling such as Amazon Athena might come in handy.

To make complying easier, you might want to insist that software developers take steps with their logging frameworks to remove personal data from log events if the events are not necessary. Bear in mind that under GDPR rules, device identifiers, IP addresses, postcodes, and so forth could be considered personal data since they could be used to single out an individual, so consider those too.

Backups

It's very likely that you'll have personal data in your backups. The GDPR may therefore impact your retention policies. Under the right to erasure, a data subject can ask you to remove data about them. If you only delete that subject's data from your production systems, you'll still have copies in your backups.

You'll want to ask a friendly lawyer about this, but from my own research into the subject, it looks like it should be reasonable to remove data from production databases and inform the data subject that whilst their data will still exist in backups, these will age out in 30 days, or whatever, according to your retention policy.

In the event that you need to restore from backups, you'd need to erase that data subject's data again, so erased subjects would need tracking, at least for the length of your retention policy.

Creating a "backup administrator" role, preventing access to backups by anyone else, and limiting the number of individuals who have that role will help reduce the number of individuals who have access to erased data during the backup retention period, which seems like a reasonable measure to take.

Dev/test datasets

Some companies are accustomed to being able to restore copies of production data into staging or development systems in order to facilitate testing. There may be an argument to allow this in staging environments, assuming access to those is limited in the same way as production access. However, allowing all your developers access to your full dataset is definitely a no-no under the GDPR.

Commercial solutions (such as Data Masker from Redgate Software) can take a dataset and mask sensitive data as part of an ETL operation into another database. I›ve also seen organisations attempt to build these themselves.

It may also be sufficient to generate dummy datasets for use in development environments, and tools can facilitate this process. You'll need to ensure your generated data is of a realistic size and cardinality, otherwise your dev systems will perform very differently.

Which of these approaches is the most appropriate for you depends on workload. Close collaboration with development teams will be important here.

Monitoring and alerting

Articles 33 and 34 require that in the event of a data breach, you notify affected data subjects, along with the supervisory authority — there will be one of these for each member state in the EU.

Obviously, this is only possible if you know you've been breached, and so monitoring, security scanning, and alerting will all play their part here. Generally, I'm a fan of open-source solutions, but in the case of security monitoring, turning to vendors is the smart option since they have whole teams of people working to keep their solutions up to date with details of the latest threats.

Web application firewalls (WAF) can help mitigate common modes of attack on web applications and APIs, watching requests for the fingerprints of these sorts of attacks and blocking them at their source. For example, a WAF might scan the content of every HTTP request, applying a list of regular expressions that match known SQL injection attacks. In the event that a pattern matches, the request is blocked instead of being forwarded to the application cluster behind it.

Scanning outbound network traffic can help identify data breaches — smart modern tools like Darktrace use machine learning to build a model of what normal looks like, in order to look for anomalies and apply pattern matching to typically "personal" data such as credit cards, postcodes and e-mail addresses. Seeing too much of such data leave your network can raise a red flag.

Inside the network, intrusion-detection tooling can help identify when your systems have been accessed by bad actors, either by scanning network traffic or by watching log data. Alert Logic and Threat Stack both have offerings in this space, and Amazon GuardDuty offers some of these features too.

Other good hygiene

There are a couple of other security practices that you should be employing here.

You should encrypt everything in transit and at rest. There's really no reason not to do it these days — and cloud vendors even provide primitives to make this easier for you. It's more straightforward to design this into an infrastructure from the beginning than to retrofit it, so make sure it's a design consideration up front.

Network design is still important. Protect your perimeter, and use host firewalls and security groups inside the network to limit access to your systems. Separate your management tools from your other systems, and keep each environment distinct from the others to prevent leakage of data between them. In some cases, it may be appropriate to segregate different classes of data into distinct databases, in order to further limit access to them on the network.

It should go without saying, but you need to ensure you have a regular software-patching regime — and not just for your infrastructure components. Keep on top of newer versions of your application dependencies, too — and employ tools like Snyk in your CI pipeline to get alerts on dependency vulnerabilities before your code makes it into production. High-profile incidents such as the Equifax data breach are often the result of insecure libraries still being in use.

Security is as much a development concern as an operational one. Tools like OWASP ZAP and Gauntlt can look for security problems in application code before they go live and cause trouble.

Consider what external services you make use of, and what data you pass to them. If you're using SaaS logging providers, for example, be aware that you may be passing personal data outside of your network, and that those providers then also have obligations to your data subjects.

Conclusion

The GDPR is an unavoidable fact of life for anyone working with data about EU citizens. Taking care of this personal data is an organisation-wide responsibility, but the operations part of the business can provide a lot of supporting tools to help deal with the multiple facets of this problem.

The GDPR doesn't substantially extend the provisions of the DPD, and so a lot of what I've described here is good practice that you should already be following. The penalties for not complying with the GDPR are much higher, however. It's time to stop looking at security as an obligation, and start making customer's data privacy a reality.

Read online on InfoQ

KEY TAKEAWAYS

The GDPR will fundamentally alter the way global organizations collect and manage their data.

Violating the regulation could result in fines of up to 4% of global revenue for your organization.

Key requirements of the GDPR revolve around managing the way data is collected, maintaining visibility into how that data is used, and enforcing restrictions on data use.

New tools, frameworks, and ways of thinking about data management are going to be required to pass the basic GDPR test and avoid violating the regulation.

Ultimately, the GDPR presents an opportunity to modernize your data-management strategy and empower your data-science programs.

What Do Data Scientists and Data Engineers Need to Know About the GDPR?

Data management is about to get a lot more difficult for global organizations, thanks to new privacy regulations in the EU. These new regulations will have far-reaching effects on any programs that use data at scale.

Specifically, the EU's General Data Protection Regulation (GDPR) will come into force on 25 May 2018. And with fines of up to 4% of global revenue, the GDPR is the most consequential data regulation to be found anywhere in the world.

While the GDPR theoretically applies only to EU "personal data", the regulation outlines this as any data that could lead to the

identification of a person. In practice, this means that any EU data at scale should theoretically fall under the purview of the GDPR, as study after study has shown that enough data of nearly any kind can shed light on the individuals who generated it. To pick just one example, a group of researchers recently demonstrated that aggregate cellular location data (such as the number of users covered by a cellular tower at a specific timestamp) — which, in theory, sounds like it should be anonymous — can actually identify an individual's trajectory with 73% to 91% accuracy.

So, what should data scientists and data engineers — the people responsible for collecting, organizing, and using data within organizations — think about the GDPR? How should they design their data strategies?

What you need to know about the GDPR

The GDPR creates legal requirements that fall into three basic buckets: collection management, data visibility, and restrictions on data use.

Collection management involves managing the data that organizations gather and the ways they are collected. The GDPR mandates that privacy be prioritized — at the time of data collection, for example — with restrictions on data tied to the consent of the data subject, meaning the data subject will frequently have to understand and agree to whatever your organization wants to do with their data. This means that when your organization collects data that a EU subject generates, understanding exactly why your organization is collecting that data and tagging that data at the time of collection is going to be paramount. (More on this below.)

Data visibility means understanding what data your organization has and how long you've had it (and how long you plan to keep it). By now, most organizations understand that data is the "new oil" and many are doing their best to collect as much data as possible. But most of those organizations don't fully understand the data they have or where they're storing it or its provenance once it's been stored.

We at Immuta frequently come across this as a combination of compliance and IT architecture issues, with data silos and different teams and database administrators responsible for a wide variety of data and no single source of truth. With GDPR requirements in place, this level of variation can't be the norm. If a user asks you to delete their data — often known as the "right to be forgotten" — your organization will have to know where their data is and then delete it. Examples of this type of visibility requirement abound within the GDPR.

Lastly, and perhaps most importantly, restrictions on data use mean that your organization is going to have to enforce purpose-based restrictions on data. If a user only consents to "marketing" as a purpose for their data, for example, you're going to need a way to track and enforce that restriction all the way from collection to use. The GDPR lists six broad purposes that are acceptable, and each organization is going to refine its own list of what purposes its legal departments deem compliant with the GDPR. This guide, for example, suggests having only 15 purposes for data across an entire organization. Tracking these purposes — and proving that data with certain purpose restrictions has only ever been used for that reason — is going to be one of the most important and difficult requirements of the GDPR in practice.

How to pass the basic GDPR test

Imagine the GDPR is already upon us, and data-protection authorities across the EU are enforcing the regulation.

At the moment of writing this article, it's clear that many of the GDPR requirements are still relatively ambiguous, and the regulators will engage in much fine-tuning over the following months, if not years. This means that, in all likelihood, regulators won't be expecting 100% compliance with the GDPR the day it goes into effect. Rather, they'll be expecting a reasonable, serious effort to comply with the regulation's major tenets.

What does passing the basic GDPR test mean?

Organizations will need to be able to demonstrate compliance with each of the buckets outlined above — understanding the data they have, when they collect it, what they've used that data for — and be able to prove all of this to regulators or data subjects, who may be entitled to reports illustrating compliance with these requirements.

From a practical standpoint, this means that, at a minimum, every piece of data that your organization collects is going to need new metadata with the fields "purpose" and "time of collection". This way, you'll be able to track and enforce restrictions on its use and you'll be able to enforce policies on data retention, meaning you'll delete or anonymize that data after a certain period of time.

If you can demonstrate that, at every point from data collection

> Creating a holistic data strategy, and a centralized place for data management across your organization, will finally allow data scientists to do what they're the best at.

to data usage and deletion, you understand exactly what data you have, how long you've had it (and plan on keeping it), and what purposes it's been used for — and that each of these buckets are in keeping with GDPR requirements — your data-management program will likely pass the basic GDPR test with flying colors.

The GDPR opportunity

All that said, smart organizations will see the GDPR as more than a new set of demands. Agile, data-driven organizations will see the GDPR as a true opportunity to rethink the way they approach their entire framework for gathering and using data.

The key differentiator of tech giants like Amazon.com or Google is how calculated they are about the data they gather and use. This is not a post hoc operation, but one based on careful planning and engineering. Having the right data is what allows them to disrupt verticals from marketing to retail to grocery stores and more.

Indeed, academic literature has long demonstrated that good governance translates to better performance. The same can be said about data management. Better, longer-lasting data-driven insights will require more deliberate thought and planning into how data is collected, and what data an organization has at its disposal.

In fact, if there's one major opportunity presented by the GDPR, it's to finally give data scientists a centralized understanding of what data they can access and use. I constantly see that the title "data scientist" is, in practice, more akin to "data scavenger", where a good deal of a data scientist's time is spent trying to find the data they need, then to get access it, then to transform it into the right state, rather than simply using it.

This process leads to huge amounts of time wasted and potential lost. Data scientists aren't hired to scavenge for data, or to create one-off, per-project solutions to gaps in their organization's data strategy. Data scientists are there to turn data into insight. That's what they are good at — and that's why they're frequently so expensive.

Creating a holistic data strategy and a centralized place for data management across your organization will finally allow data scientists to do what they're the best at — and will help your company move faster, becoming more efficient and more adaptable in the process.

What comes after the GDPR?

Beyond the immediate opportunity presented by the GDPR lies an entirely new way of thinking about data, one that is going to become increasingly important as new regulations on data emerge. Indeed, from Turkey to China and elsewhere, data is becoming more and more regulated, meaning that data management is going to be one of the most important enablers for data-driven organizations and one of its biggest challenges.

Here are a few insights about the future of data management:

- There's no such thing as a data lake. Often when it comes to data management, an organization's first instinct is to think that putting all its data in one place will solve every problem it has. When it comes to data lakes for processing purposes

(like Spark), this makes a lot of sense. But for governance and data discovery, data lakes frequently create huge problems, quickly turning into data ponds and then data swamps as new data is added, new tools for data storage emerge, and underlying IT architecture evolves. Thinking you'll solve your data-management problems by centralizing where you store your data is a recipe for long-term troubles.

- Diversity is your friend. Instead of attempting to standardize the way you store your organization's data, which can be nearly impossible in large organizations, I recommend thinking about the long-term adaptability of your approach to data management. That is, assume that you're going to have diversity across your storage systems and data-science tools — indeed, this diversity is inevitable. Once you realize that standardizing where or how your data is stored is not your number-one priority, you can move on to thinking about how to enforce and support policies with respect to that data, which is the backbone of any data-management strategy.
- Audit. Audit. Audit. If you can't audit, you can't prove that your data-management framework is working and you can't demonstrate that to regulators. Ensuring that there's a centralized ability to audit and to create audit reports is going to be a key component of any data-management strategy. And make sure to test your audit abilities before they're needed. Organizations frequently think they're collecting the right data for their audit needs, and all too commonly learn about log errors only once it's too late.

There are, of course, many more key tenets to a future data-management framework for the GDPR. But the major takeaway for your organization should be that data management can no longer be an incidental component of your data strategy, in the IT department or otherwise. The increasing importance of data science across organizations, combined with the rise in regulations on data, means that organizations will need to prioritize data management more and more.

PREVIOUS ISSUES

59

Microservices - Patterns and Practices

While the underlying technology and patterns are certainly interesting, microservices have always been about helping development teams be more productive. Experts who spoke about microservices at QCon SF 2017 did not simply talk about the technical details of microservices, but included a focus on the business side and more human-oriented aspects of developing distributed software systems.

Observability

58

This eMag explores the topic of observability in-depth, covering the role of the "three pillars of observability" -- monitoring, logging, and distributed tracing -- and relates these topics to designing and operating software systems based around modern architectural styles like microservices and serverless.

Streaming Architecture

57

This InfoQ emag aims to introduce you to core stream processing concepts like the log, the dataflow model, and implementing fault-tolerant streaming systems.

Faster, Smarter DevOps

56

This DevOps eMag has a broader setting than previous editions. You might, rightfully, ask "what does faster, smarter DevOps mean?". Put simply, any and all approaches to DevOps adoption that uncover important mechanisms or thought processes that might otherwise get submerged by the more straightforward (but equally important) automation and tooling aspects.