DigitalRoute

Your guide to moving to a usage-based business model

Let's get under the hood of usage-based pricing

Welcome to the usage-based economy.

Customers don't want to own anymore – they want to *use*.

And that's in all parts of their lives: how they use software at work, how they listen to music, even how they drive cars.

It goes for businesses too. Organizations don't necessarily want to own all the equipment and infrastructure they need to run. They want to be offered the *service* instead.

By selling outcomes and not ownership, you let customers pay for the value they need. Which is good for your customers – and *excellent* for you: users grow, but usage scales.

A usage-based business model ensures your offering and operations can respond dynamically to changes in the market and changes in customer use. So you can grow your revenue, adapt to the ebb and flow of demand, and explore new opportunities.

Plus, usage-based businesses are often valued higher, because they can demonstrate their products are actually used.

Usage-based SaaS businesses are valued substantially higher: a 21.6x revenue multiple compared to 14.4x.[1]

1. The Usage-Based Pricing Playbook, OpenView, January 2021.

The Trend

But all this is easier said than done.

There's a *lot* to do: setting an initial strategy and pricing, creating a go-to-market strategy, and updating finance and operations.

You'll need to figure out all sorts of things – like which usage-based metric to use, your subscription or pricing model, how to process data about usage, and how to evolve your usage-based services as you grow.

This guide zooms in on one of the trickier parts of monetizing usage-based business models: how to handle the usage data that generates your revenue.

We've spoken to a bunch of people who've spent their careers solving this problem. And we've compiled all their hard-won advice and experience into this guide.

Ready?

The Journey

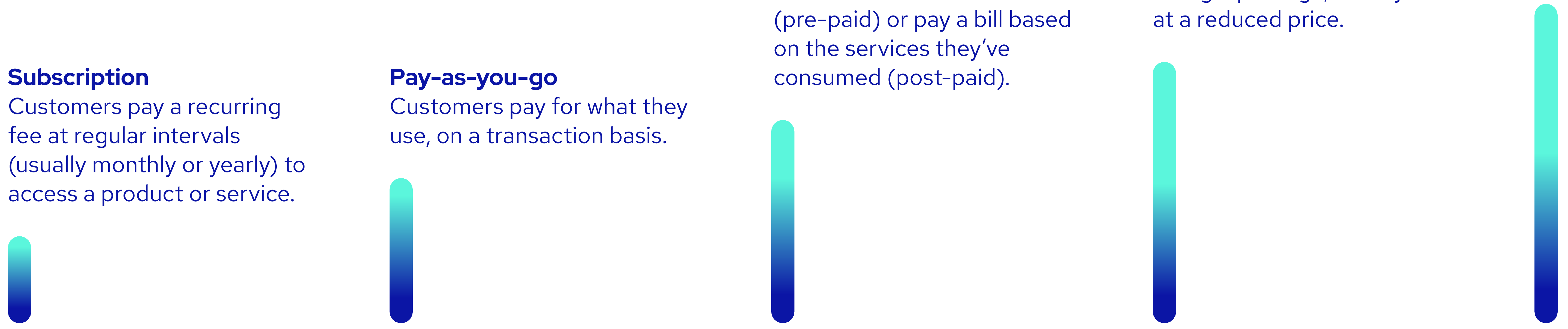
Contents

Chapter 01

How to define usage.

01

How to define usage.



Subscription
Customers pay a recurring fee at regular intervals (usually monthly or yearly) to access a product or service.

Pay-as-you-go
Customers pay for what they use, on a transaction basis.

Pre/post-paid
Customers either purchase a service allowance in advance (pre-paid) or pay a bill based on the services they've consumed (post-paid).

Shared bundles
Bundling combines several services or products into a single package, usually at a reduced price.

B2B2X and the partner ecosystem
Here B2B and B2C converge: it's B2B to any end user. Partners team up to offer innovative services and drive new revenue.

Common usage-based business models

Before you start making any changes, you need to pick the right usage-based pricing model. And this involves a fair bit of experimentation: there's a definite sweet spot between "allowing for growth" and "ensuring the stability of recurring revenue".

A flat-rate subscription will box you in. Once you've scaled your number of users – what next?

No, instead you should look towards a *hybrid* model. This lets you offer a low subscription offer to lock in loyalty and establish that recurring revenue – while giving you the freedom to achieve growth with additional usage-based services on top.

Once you've chosen the right model, you need to choose the right *metric* to determine what customers will pay. This needs to capture and communicate your product or service's value, while also being flexible, scalable, predictable, and feasible to monitor.

And then?

It's time to understand how that usage turns into revenue.

Quick time to market for new VMware cloud offerings

VMware was looking for a centralized billing system for subscription and pay-as-you-go business models. As part of the solution, DigitalRoute's Usage Engine was deployed to capture and refine cloud infrastructure related events for consumption billing purposes.

Watch the webinar

Turning usage into revenue.

Turning usage into revenue.

Hard-won tip:
Don't be fooled into thinking this is a case of simply lifting and shifting your product portfolio onto an all-you-can-eat subscription model.

First things first, there's nothing simple about it. Even an all-you-can-eat model requires understanding who your customers are, what they're consuming and then *transparently* communicating that back to them. (All in real-time.)

More importantly though, this model doesn't scale. With all-you-can-eat services, you're basically saying that all of your services are free once someone subscribes. (Yeah, even new services.) Instead, you need to make sure you can charge for *all* the services you offer.)

And this is especially true for B2B organizations.

For a seriously scalable usage-based business model, you need different levels of subscriptions. That way, you can incrementally increase revenue as you deliver more value to customers, allowing them to start small and grow with you.

But this is where it starts to get tricky.

Think about it: you'll have several subscription types with varying service levels, and each with different costs and rules for service limits, then multiplied by the number of customers.

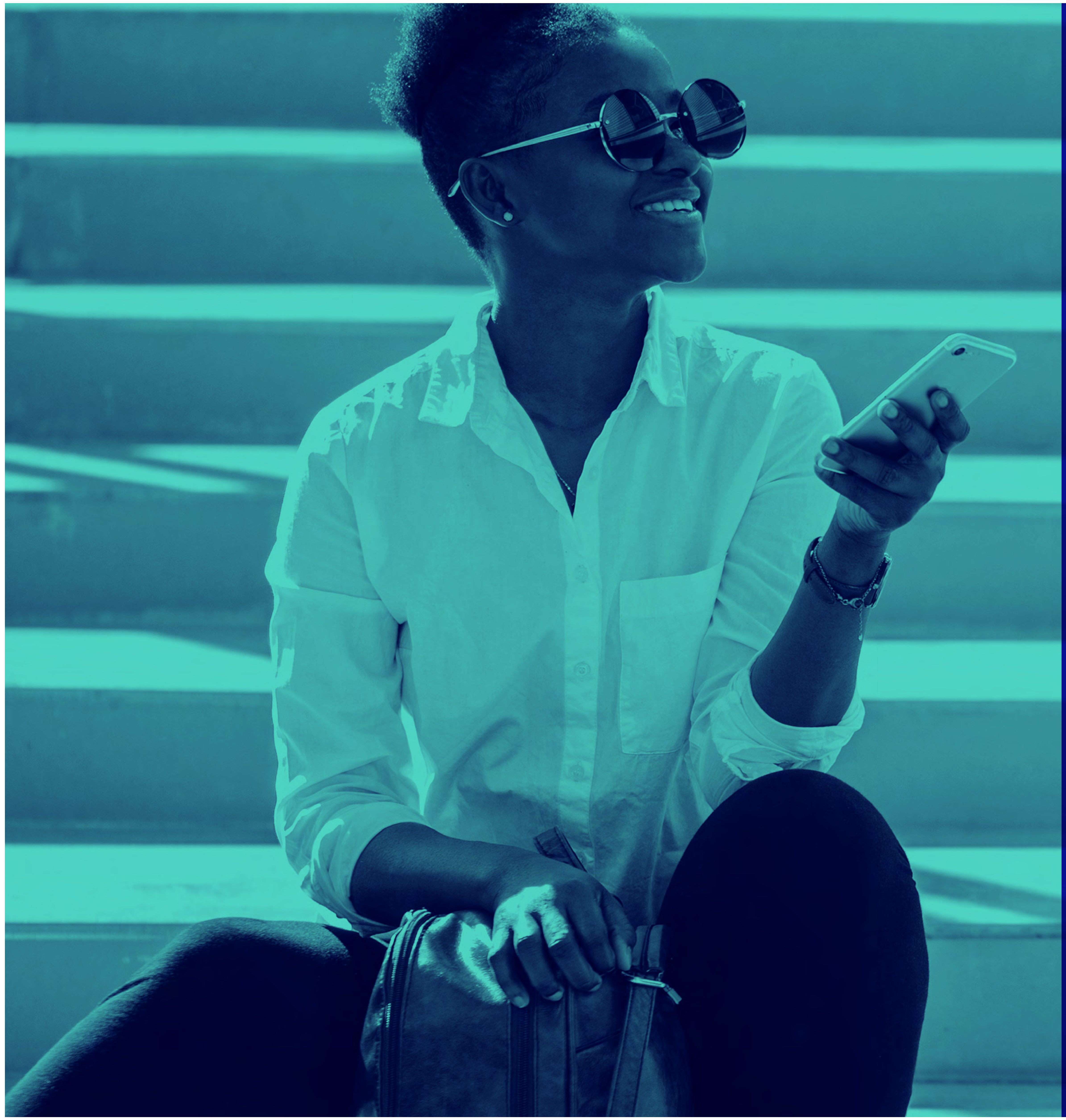
All of this means that you need to understand – in detail – how subscriptions are used.

With this level of complexity, it's not uncommon to have tens of thousands of usage events per second hammering your systems.

Not once or twice a day.
Every. *Second.*

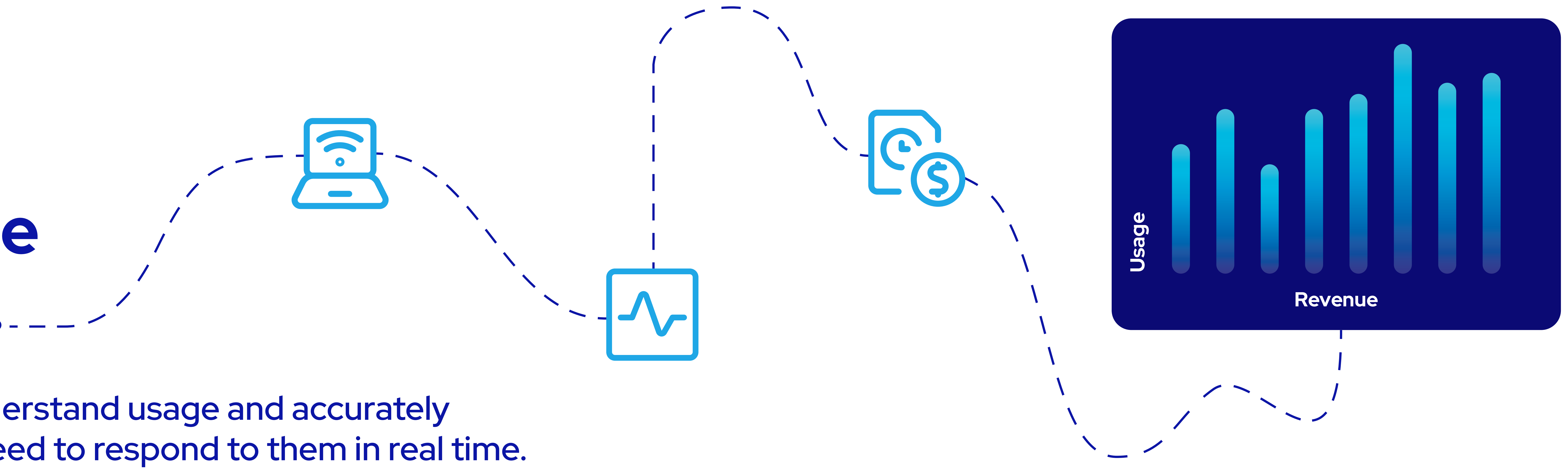
The worst-case scenario for usage-based billing: revenue leakage

The data that tells you about how people use your product is critical to your revenue. If you lose any of that data, you lose revenue. In fact, according to EY, companies can lose 1-5% of their revenue due to errors in data processing.



Not once or twice a day. Every. Second.

Turning usage into revenue.



These events help you understand usage and accurately record revenue, and you need to respond to them in real time.

Which involves...

- Processing the data to compare it to the subscriptions and packages customers have signed up for.
- Using the data to measure your business model's performance.
- Triggering customer interactions to alert them as they approach their usage limits, to upsell higher-value packages, and intervene to mitigate a poor experience.
- Figuring out how much each partner should get paid (if you offer joint services).

At a minimum, this touches CRM (customers and contracts), CPQ (products), billing, EMS (entitlements), and ERP (billable items for revenue recognition and partner settlements).

So that's most of your quote-to-cash stack, then. (And when we say stack, we mean loose confederation of systems not *exactly* famous for playing well together.)

This stack simply wasn't made for usage-based billing and customer journey orchestration, with their relentless, always-on cycle of collecting data, categorizing it and taking action.

(Even if it wasn't made that long ago.)

Rethinking the order-to-cash process in public transportation

The main passenger railway operator in the Netherlands was looking for a centralized billing and invoicing system to quickly add new travel products. As part of the solution, DigitalRoute's Usage Engine was deployed to capture and refine travel related events for consumption billing purposes.

Watch the webinar

Chapter 03

The foundations of solving the problem.

03

The foundations of solving the problem.

Here's what you don't need: a new billing system. (Not yet, anyway.)

We see it a lot.

Company X jumps in at the application layer, eager to revamp their revenue systems. The RFP goes out, the bids come in. If they're lucky, one of those bids flags an overlooked and underappreciated problem – data – and the giant tender for the giant contract delivering a giant system stops there.

If they're not so lucky, they end up with a gleaming Rolls Royce of a system they can't drive because, while the engine is big, it's underpowered. The data going in is outdated, dirty, incomplete – or just the wrong sort.

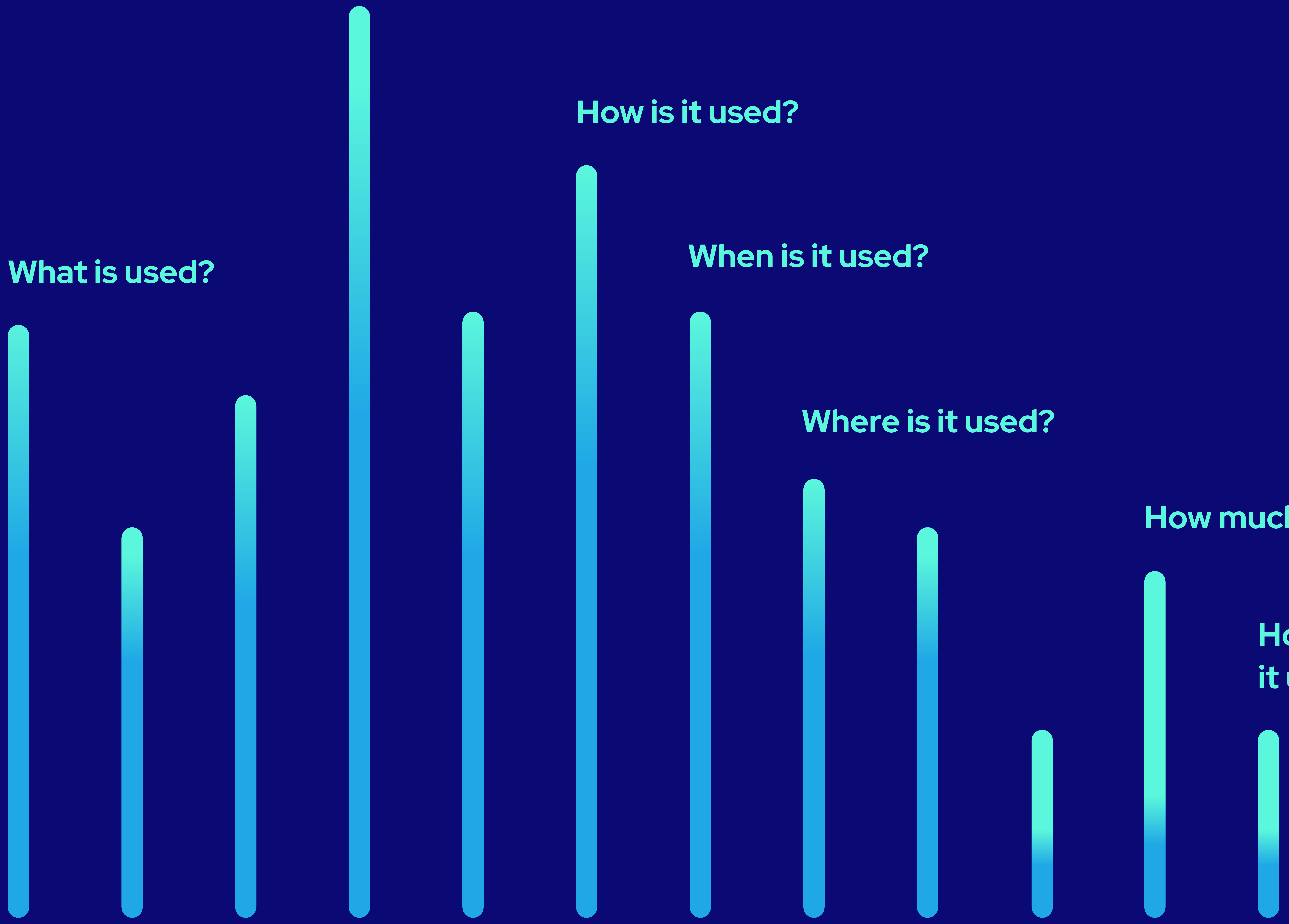
Facing the data challenge *first* will save a lot of pain.

That starts with figuring out what usage data you have and where it lives – which can be tricky. Enterprises today have data in many disparate systems.

What's usage data?
In its simplest form, usage data shows you how your service or product is used.

What's usage data?

It tells you things like:



What is used?

Who uses it?

How is it used?

When is it used?

Where is it used?

How much is it used?

How long is it used for?

Examples of usage data include: unlocking a shared bike, calling the API of a cloud service, swiping a card on public transport, clicking play on an online movie, a toll camera identifying a license plate.

The foundations of solving the problem.

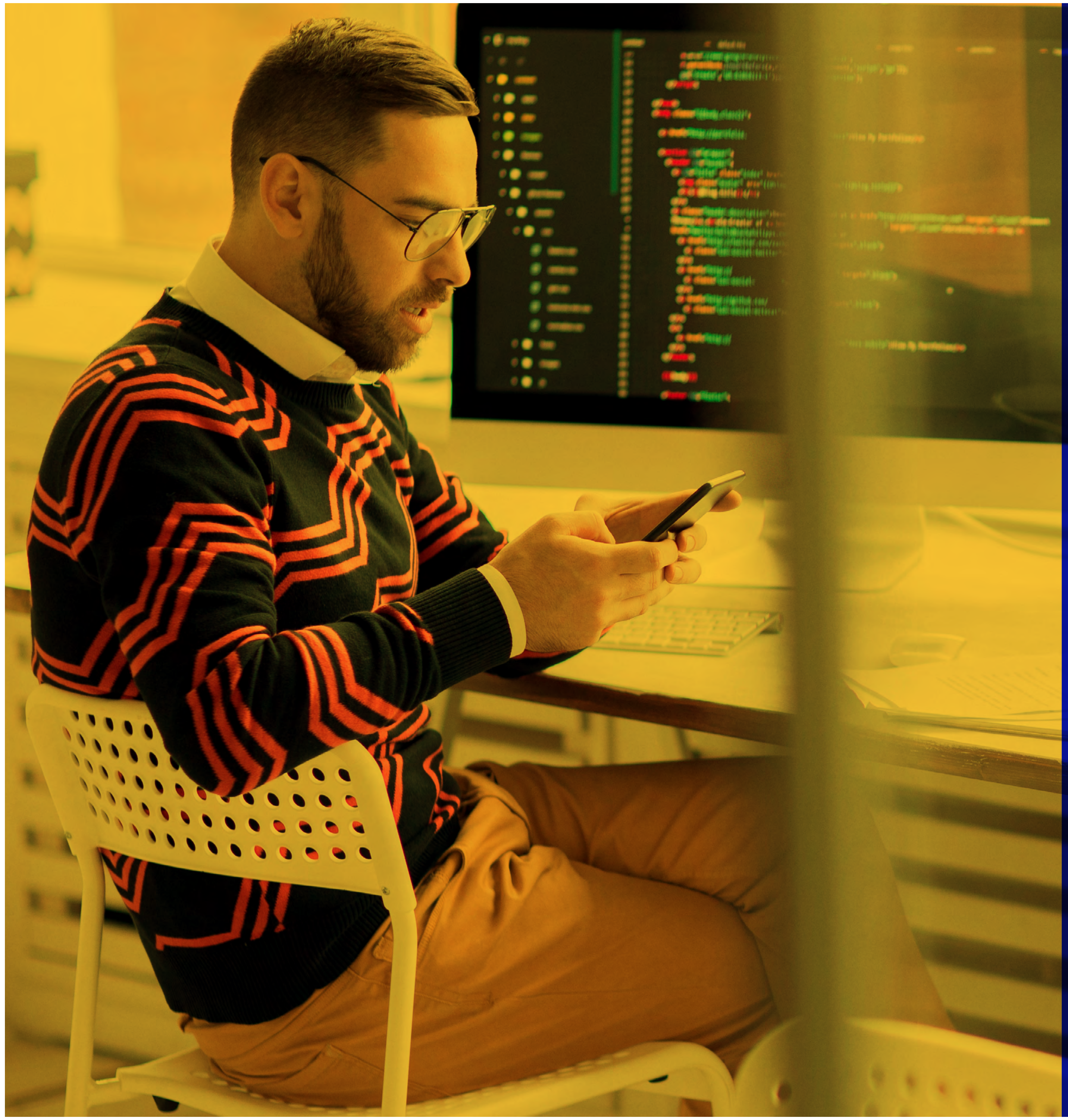
Once found, you need to ensure that data is accurate and timely. Then you need to integrate it with revenue and billing systems to deliver one single, transparent invoice, and a simple overall experience for customers.

And this is a *major* point. In a usage-based model, billing becomes a fundamental part of your customer experience. Get it wrong (and send inaccurate, outdated bills), and you risk losing your customer's trust – and potentially their account.

Reckon it's time to speak to an expert?

Read on and you'll find a few different ways you can solve this problem, avoid potential revenue leak, and succeed with your new business model. But if you'd rather speak to an expert first, we get it.

Contact us here for a chat



Start with figuring out what usage data you have and where it lives.

We don’t want to get technical, but we think it’s important to know what that looks like under the hood.

1

Data is taken from dozens of product and service systems and normalized so its format is consistent between each system (and usable).

2

That data is aggregated into the billing period, enriched with data from other systems, such as CRM (order data), and then bound to the customer profile.

3

If it’s incomplete or potentially unsound, then you need an automated way to go back and find what’s missing. (Any changes need to be auditable too).

4

Only once the data is complete can your business logic be applied, by checking the usage record against customer entitlements, for example.

5

And then – finally – your billing, auditing, marketing and BI systems register it as a revenue-generating event.

Pretty tough, huh?

Most revenue systems aren’t built to connect and monetize data in this way, at this volume, or at this complexity. These traditional finance-led IT stacks can’t support launching new digital services, which are fast to market, highly changeable, and super responsive.

The foundations of solving the problem.

Instead, you need to add a brain to your billing system – a software solution that:

- Mediates between systems.
- Normalizes their data output.
- Solves for usage data management.
- Handles business logic in robust workflows.

With a software like that, you can tighten up your quote-to-cash stack to make it more agile. You can also disentangle the arduous, manual, error-prone workflows that everyone has (but few admit to), which are *even less* compatible with usage-based billing.

(We've seen it all, including a number of Fortune 500 companies whose master financial data records depended on a bunch of *terrifying* Excel spreadsheets.)

To be clear, we're *not* talking about application integration. This software should work within your existing quote-to-cash process, connecting old, new, and not-yet-thought-of IT infrastructure – without creating more silos.

Bringing partner applications to the edge with Google Cloud

DigitalRoute is providing its solution on Google Cloud to collect usage information from edge services and network elements and prepare it for billing, partner settlement or any other form of monetization.

Read more

In summary, life without a usage data layer looks like:

1. Huge revenue leakage, as usage events get lost or dropped by unstable or unconnected systems – or by human error during manual processes.

2. Multiple billing processes and systems that make it impossible to send real-time, accurate invoices to customers.

3. Duplicate systems (and middleware) that drive high operational costs and create errors.

4. Sluggish product launches and slow reactions to changes in the market.

5. Poor billing experiences.

Build, buy, or integrate?

04

Build, buy, or integrate?

Hard-won tip:
Don't forget to consider the cost of delay that comes with using a generic integration software or building your own system. Slower time-to-market, inflexible structures that can't adapt to changing demand, and bad customer experiences are all a real risk.

There's three main ways to capture, process, and monetize your usage data:

1. **Application integrations**
2. DIY custom builds
3. Purpose-built software

Application integrations

A traditional integration solution or "middleware" is the first thing IT will want to explore. It's not a bad shout: these solutions are built by smart people at companies with excellent track records for integrating complex systems.

But they weren't built for handling data that leads to revenue.

This is a many-to-many challenge. The vast majority of middleware is designed to connect one source to another (or one source to many sources). But with usage data, you need to connect many sources to many sources.

Achieving this with integrations means using a *significant* amount of middleware. And with that comes more complexity, more silos, and less efficiency. Your stack gets harder to maintain and difficult to adapt.

And in a usage-based business model, finance needs transparency into *exactly* what's going on. As billing becomes an increasingly important part of your customer experience, it's vital to get it right. Customers need to trust their bill, gain access to usage in real time, and see consistent amounts across systems.

Ultimately, this data is intrinsically tied to your revenue. So it can't be left *only* to IT.

Build, buy, or integrate?

There's three main ways to capture, process, and monetize your usage data:

1. Application integrations
2. **DIY custom builds**
3. A purpose-built software

DIY custom builds

No approach offers quite as much control over what the system looks like and what it does as doing it yourself.

Unfortunately, that sense of control evaporates once new requirements are added or the scale changes.

These are complex, changeable workloads and the applications handling them must be *seriously* robust. (At the end of the day, revenue depends on them.)

It's almost always messy work. You need to be able to iterate quickly without having to go back and fix everything you've already done each time you want to make a change.

Plus, you need to consider the costs. Building your own system requires a dedicated team that can continuously hack away at your data layer to keep up with changing demand and needs.

Which brings us on to purpose-built software...

Your best bet is the Usage Engine.

Your best bet is the Usage Engine.

We purpose built the Usage Engine because, when it comes to revenue, you simply can't wing it.

Built specifically to solve the challenges enterprises face when adopting a usage-based business model, it finds, collects, normalizes, and corrects *all* of the data you need to generate revenue. By the time that data hits your ERP systems, it's reliable and ready to drive invoices, insights, and intuitive experiences.

It's the turbocharger for that Rolls Royce we talked about earlier.

SAP IT on SAP Convergent Mediation by DigitalRoute

SAP Convergent Mediation by DigitalRoute helps to capture and process all data about how customers are using the services.

Watch the video

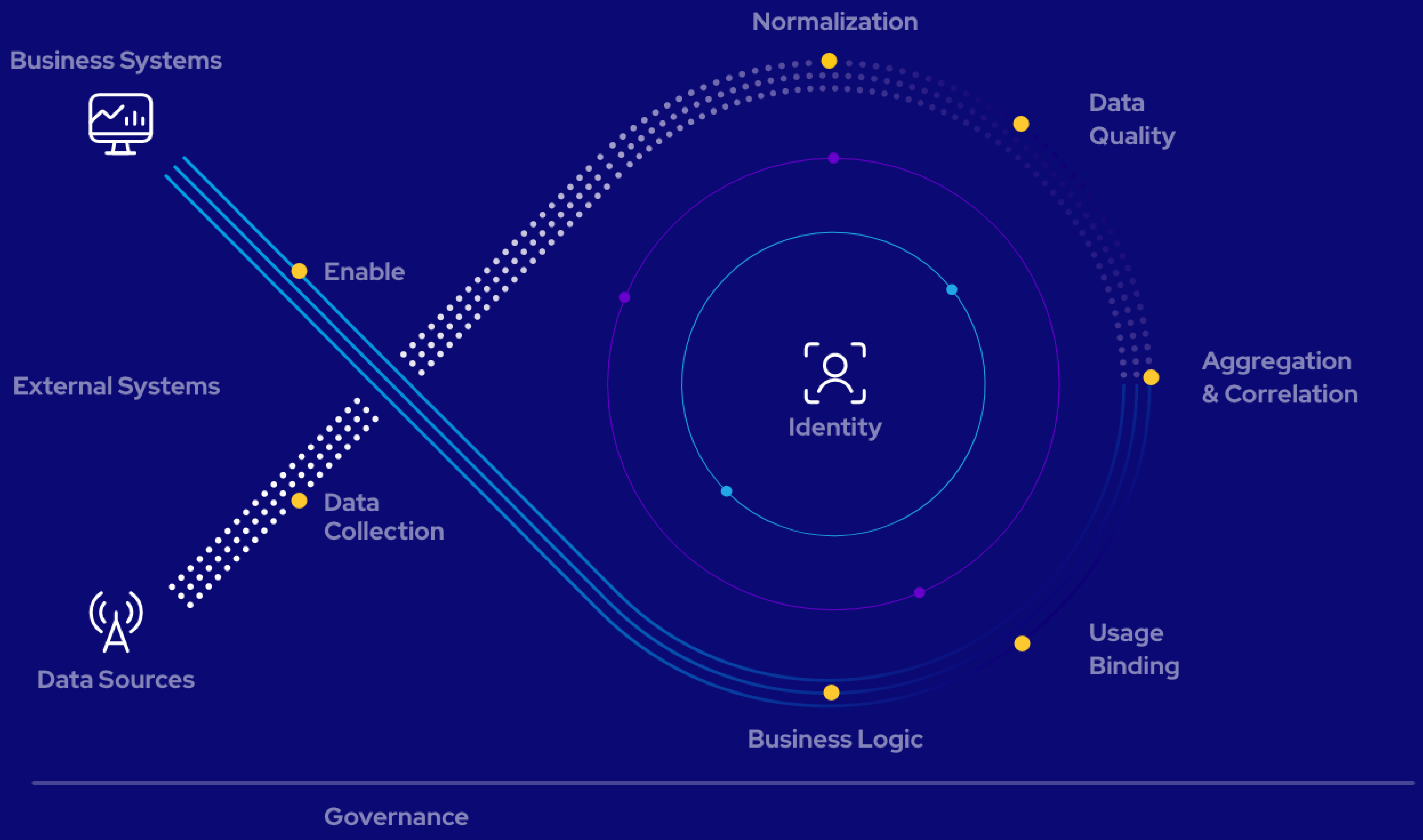
It's the turbocharger for that Rolls Royce we talked about earlier.

With the Usage Engine, you can:

- Prevent revenue leakage
- Tailor user experiences
- Create new business models
- Understand service consumption
- Streamline partner settlements
- Improve billing efficiency

See all solutions

Your best bet is the Usage Engine.



This is how it works, step by step:

- Data collection: Usage data is collected from any system (usually several, possibly hundreds).
- Normalization: Data is transformed into a consistent and usable format.
- Data quality: Data is cleaned, and errors are identified and corrected so none is lost.
- Aggregations and correlation: All the data is combined and enriched to be matched with other data sources, such as a CRM or CPQ system.
- Usage binding: The usage data is bound to an identity, user or asset (a critical step for usage-based models).
- Business logic: Now the data is processed against customer contracts and entitlements.
- Enable: Here, the data hits your revenue systems, driving the billing cycles and customer experiences that underpin your usage-based transformation.
- Governance: The software creates an auditable record for all the data processed, so you can go back and see exactly what's happened.
- External systems: Rather than a one-way flow of data, the software can connect to your other systems (like a CRM) and send data back to them.

Learn more

The four principles behind the Usage Engine

Adaptive

Agnostic

Accurate

Automati

1

It adapts to any future thanks to its low-code configurability, built-in data integration catalogs, and widely tested system interoperability.

2

It's technology agnostic, keeping the peace between systems.

3

It's accurate for every moment of usage, so you can monetize your data correctly and with full transparency for your customers.

4

It automates quote-to-cash workflows at scale.

Your questions, answered.

06

Your questions, answered.

When customer satisfaction and revenues are at stake, your solution's automation engine, data algorithms and logic engines must be reliable, robust and stress-tested in the real world.

Sometimes it's hard to know where to start. We can help you identify the right business model for you and your organization: we've seen it all when it comes to usage-based billing, and have been here since the start.

(Plus, our software has been battle proven in the most demanding environments.)

If you've got something specific in mind that you want to talk about, or you want to hear more about how our solution could help your organization, then get in touch.

Contact our experts

DigitalRoute