FUUZ
ignite innovation™

WHITE PAPER

Introducing the Fuuz Platform

BY CRAIG SCOTT
FOUNDER AND CEO, MFGx

FUUZ PLATFORM
Contents

Back to Contents

FUUZ PLATFORM
Introduction

The Fuuz Platform can be difficult to understand because there's nothing quite like it on the market. At its core, it's an anything platform as a service (xPaaS) solution that is truly a platform in a box, ready to be used on Day 1. The Fuuz Platform allows developers and anyone within an organization, regardless of technical ability, to deploy pre-built apps, develop custom apps, design screens and implement other processes that connect people, processes, machines and data across small to large enterprises.

That might sound like a big promise but it's the truth – and we're going to tell you why in this white paper. In this piece, you'll discover the four components of a platform and take a deep dive into the Fuuz Platform's tech stack where the magic happens.

We'll also explore how our multi-tenant, multi-enterprise architecture; powerful cloud computing system; no-code, low-code and pro-code tools; and features like the Browser Extension and Fuuz Gateway give users unparalleled control over their business processes while making security a top priority.

Back to Contents

FUUZ PLATFORM
What is a Platform?

A software platform, or a rapid development platform, is a technology that enables administrators to do many things that cannot be done with monolithic applications.

When you think of a monolithic application, you might think of something you would install on your smart phone or device. These types of applications are generally very focused on their purpose, industry and functionality. They are simple by nature and they are relatively unchangeable. As with most applications of this variety, there may be a settings or setup option providing you with a handful of binary choices you can make about how you want your experience with that application to be. Ultimately, your experience with that monolithic application is dictated by the software developer and the way they feel most of their customers will be utilizing it.

The platform concept enables the consumer to have much more control over how they will use or interact with the application. The concept of a platform is not new; applications like SAP, Oracle, NetSuite, MS Dynamics and Salesforce.com are all platforms. There may be some specific settings or options that can be configured quickly and users can usually customize the solution further, if desired. In a platform, this is accomplished by going into one or more of its application layers.
The layers will be discussed here as well.

Platforms give users greater flexibility. You are not limited to just what you see on the surface of an application, or what you can control with a few binary options provided to you by the software developer. With some platforms, this level of flexibility is minor, in that you can add or remove fields from the user interface. Other platforms go further and give users control over the business layer, which is where all of the functional logic is handled. This level of autonomy provides the customer with the ability to change how the end users interact with the application, as well as what happens under the hood from a data and transactional standpoint.

Some platforms will even allow access to the data layer, which allows users to personalize the data that is stored and retrieved from the system. The above statements do not apply to monolithic applications. By virtue, the software developer of a monolithic application does not want, nor can they afford, to have their customers manipulate this level of the system. There are clearly pros and cons to both approaches.

Platforms simplify the process of developing applications for the consumer. This is achieved with four main components that are largely abstracted from a myriad of technology lying beneath each, the architecture required to enable interaction between the components and the delivery of technology, whether on cloud or on-premise. In this white paper, I will discuss a simple 4-layer platform architecture, then draw some comparisons to how this fits within the world of Fuuz.

Back to Contents

FUUZ PLATFORM

The Software Life Cycle

As companies evolve from the startup to enterprise levels, their software needs change. This is known as the software life cycle.

At the early stage of business, software consumers are often ill-equipped to fully understand and appreciate the value of a platform. Transitioning from one phase of business development to another sounds easy, but can be a very complex process in application. The unfortunate reality is that each transition phase is painfully expensive, challenging, creates new business risks and often results in failure. Just search for "failed ERP implementations" and you'll have some fun reading.

Small and medium-sized businesses (SMBs) generally prefer monolithic applications, like Quickbooks. They are simple to deploy and maintain and relatively inexpensive to operate. There are many benefits to this, including ease of implementation, reduced cost, simple training requirements, etc. These applications are readily available; there are dozens if not hundreds that offer simple per user-based pricing, which is attractive to small businesses.

If you consider the trajectory of an SMB as they start off with a system like Quickbooks, they will eventually reach a point where the application no longer meets their needs. Perhaps they've moved beyond the $5 million to $10 million revenue environment where these simplistic systems work well. Maybe they have grown and have too many users, or now they have customers that require them to use electronic data interchange (EDI). At this point, they start to evaluate a new solution such as NetSuite, Acumatica or something in the mid-market system range.

Many organizations in the larger SMB stage start to take an ad-hoc approach to software. They have a monolithic application like Quickbooks and begin adding additional applications to address their growing needs. Suddenly, they're working with separate vendors for labeling, shipping, warehouse management and other business needs. The problem is, these applications don't communicate well and business leaders must invest a lot of time and resources in data reconciliation and audits for accuracy. This is where application consolidation starts to become attractive.

At this stage, most organizations turn to slightly more powerful software applications that are more functional than their previous solution but still only meet 80% of the company's business needs. The organization probably invests in some on-premise servers and outsourced IT because going to the cloud seems scary or risky. This perception of risk is understandable but it's just that – a perception that could create a much greater risk for their business, financials, information and even trade secrets.

There is no way that any on-premise solution can be as robust and secure as a cloud footprint with experts monitoring it every minute of every day.

Back to Contents

FUUZ PLATFORM
The Software Life Cycle

We often work with SMB owners to educate them about the difference between what appears to be a cost-effective solution and what a truly cost-effective solution is.

For example, the perception is that it's cost-effective to own a server rather than rent space in the cloud. Perhaps it is for a while – until there is a cybersecurity attack and you lose everything.

As a company grows to become an enterprise, they need to shed the skin of the mid-market software decisions they made and look for the appropriate systems to run the business. This often leads them to tools like SAP, Oracle and MS Dynamics.

These ERPs are platforms that provide scalability and robustness and allow the company to continue to grow and, as they do, develop or implement the functionality they need rather than procuring more bolt-ons. Although they cannot do everything, these platforms are designed for office-level transactions like accounting, purchasing and sales. Enterprise businesses find themselves augmenting with MES, WMS, TMS, PLM and other packages to address other areas of the business.

All of this to say that there's definitely a pathway taken by most companies as they continue to grow out of one set of software into another. Each of these transitions comes with risks, costs and implications.

Historically, only enterprises could afford a platform; something scalable, robust and in the cloud. Something that does everything they need it to do today but provides the flexibility and extensibility they need for the future. These dynamics are changing now with solutions like Fuuz.

Fuuz eliminates the need for your company to hire a team of developers, DevOps engineers and IT experts to get everything working. Fuuz is truly a platform in a box, ready to be used on Day 1.

Back to Contents

FUUZ PLATFORM

Components of a Platform

A platform consists of four layers: Data, Application Programming Interface (API), Business, and Presentation or User Interface (UI). Most platforms only give customers access to a couple of layers and a few basic customization options, but the Fuuz Platform is different – and so is our tech stack. To understand why, you first need to know what goes into what makes a platform a platform.

The Data Layer

The data layer of the platform consists of the database where the consumer of the software stores data and retrieves it for reporting or analytics. Because business logic is often dictated by data, or vice versa, having access to this is off-limits for monolithic applications. They have tried their best over the years to implement things like custom fields or attributes, but those features merely provide additional metadata about an item in the system. They cannot be manipulated or evaluated within the business logic of the software. This means you can tag some things with additional information, but you are not able to alter the way in which your users interact with the software.

The data layer is typically the lowest layer in the software stack; it's where things sit or persist until someone using the application requests access to that data or creates new data. Interaction with data can happen in a number of ways. In modern platforms, this is done via an API layer that allows read/write access to data for certain users based on their permissions or policies granted to them. This API layer is extremely important, as it is the means by which the business logic can interact with the data in your system.

There are several different types of databases available, and choosing the right one is very important when businesses are moving their operations to the cloud. That's because it's extremely challenging to revert back if you later learn that you made a design mistake.

Before there were no-SQL databases such as Mongo, there were the traditional SQL-ish relational databases. You've likely heard of Microsoft SQL, Oracle, Postgres and others. Relational databases had a time and a place, but more modern frameworks have exposed the benefits of no-SQL or non-relational databases. Most modern cloud platforms will take advantage of these document-style databases because of the flexibility and ease of use they have to offer, as well as the ability to use more modern tools for building pipelines, aggregation and query languages such as GraphQL.

Back to Contents

FUUZ PLATFORM
Components of a Platform

The Data Layer, cont.

Legacy monolithic software packages will generally use the older-style relational databases because much of the performance is inherent; there's less engineering work involved in making them work well. That said, the trade-off is that they become rigid and complex and much too difficult to be flexible longer term. With modern tools like Mongo and GraphQL, the perceived performance issues of the past are long gone; the ability to create highly relational models is no longer an issue.

The ideal database solution for your project should be a document-style database that provides ultimate flexibility long-term so you can adapt and add fields, tables and relationships without completely rewriting your application every time.

The API Layer

The term API is oftentimes used in a general sense, referring to an application programming interface. All this really means is that component of the technology allows other parts of the software application or, in some cases, third-party applications to access data.

Common terminology like Restful API, web services and ODBC are all examples of APIs most of the time. Older legacy software applications may refer to something called a "stored procedure." Stored Procedure (SPROC), is a method in which the application can perform actions to the data, oftentimes referred to as Create, Update and Delete (CRUD) mutations to the data.

Monolithic applications may combine two important layers into one as a way to save time and cost since they never planned to be extensible or infinitely flexible. Some platform architecture componentry can be easily skipped over. For example, in monolithic applications, these stored procedures will oftentimes be a combination of the API that accesses the data layer. But it will also include the business logic or functional logic that makes decisions on how to deal with data based on the user's interaction. This approach makes things very fast and easy to use – as long as you never need to extend or modify the functionality to fit your needs.

With modern platform architecture, especially those leveraging micro services, it is critical that these architectural layers of the software are kept separate. This enables scaling to address the needs of users as they grow, when transactional volume increases, or when the end user needs to adapt something to work closely to their process.

Once you've defined your data layer, you as the developer are then responsible to create the API layer that sits atop that. By doing this, you can now access the data and its elements within your business logic and presentation layers.

Back to Contents

FUUZ PLATFORM
Components of a Platform

The Business Layer

The business layer is where the magic happens for most software applications. As mentioned previously, most monolithic applications have one goal in mind and will co-mingle the business and API layers, since they really have no need or desire to be infinitely flexible. The business layer may consist of orchestrations, flows, stored procedures, scripts or other logic that is created to support the way in which transactions will be handled in the system.

As you can imagine, with a platform, not many of these things exist, as usually the platform is a blank sheet of paper for you to create something on. With monolithic applications, you will not have access to this layer except for a handful of binary choices you can make to manipulate some functionality at a very minimal level.

There are many inherent pros and cons to having access to this layer. For starters, the biggest pro is that you have full control over the way your business will utilize the application. Large-scale solutions like SAP and Oracle deliver on this promise to their customers, but implementing that scale of application is very time-consuming and costly.

One of the first cons you'll hear from someone offering you a monolithic application is that it is complex to manage and requires a great deal of skill and expertise. The solution becomes watered down to the point that you're forced to do things a certain way to provide some benefit to the end user.

The business layer is often the code or the computer language that provides functionality. When you hear about no-code or low-code, what that means is that the software company has reduced, or sometimes eliminated, the ability to have full control over the business layer. This is where many customers have tried and failed to implement such platforms; their technical teams are unable to work with such a tool in the same ways they can work with more traditional methods, such as writing C+, Python, JavaScript, etc. This is typically a big turnoff for many technical creators.

One thing that is different about Fuuz is our advanced pro-code capability. While we have no-code and low-code options for less tech-savvy users, those who need full control can certainly have it. This is a differentiator of the Fuuz platform, and one in which we take great pride in introducing to a space that has received a bad rap over the inability to build truly enterprise-grade applications.

Back to Contents

FUUZ PLATFORM
Components of a Platform

The Presentation Layer

The presentation layer, or the user interface (UI), is what many of our customers care about most. It is the only component that your business users see while the business and data layers work behind the scenes.

When it comes to monolithic software, most off-the-shelf platforms have limited UI. The provider must first create the data layer, then the APIs and finally the business layer before putting a UI together would make any sense.

With no-code and low-code software tools, providers usually begin with the UI and work their way backward because they want to appeal to the less technical users. After all, many of us can only think about what we can see or touch. If you're not a software engineer by trade, then likely you aren't going to know how to construct a proper data model, or API. All you want is a simple screen with some fields where you can enter and track data. That works really well for this purpose; this is where a system like Salesforce.com becomes very popular. It abstracted the complex stuff and provided some simple fields to be added to screens and linked together, which works really well for a customer relationship management (CRM) solution where data isn't being tracked at a very deep level.

Fuuz delivers more benefits to the customer. Our platform's data layer provides much greater flexibility in terms of deriving very complex data solutions. Our clients are typically manufacturing companies and tend to need complex data across their operations for tasks such as machine recipes, production routings and bill of materials.

Software companies have never delivered this model in a scalable and repeatable way before. That's why all off-the-shelf MES solutions available to you today are monolithic. These applications offer users a standard UI and, as long as it works for your needs, you're in luck. This UI can't be dramatically changed and that's one of many reasons customers come to Fuuz.

The Fuuz Platform helps our customers extend monolithic applications. Fuuz has the ability to connect to anything, even if that thing isn't necessarily the easiest to connect to. We have invented ways to work around the deficiencies of some other software packages in order to provide our customers with that last-mile functionality they require to operate their business efficiently and effectively.

Platform tools will generally go one of two ways with the presentation layer. The UI will either be extremely simplistic, with the ability to generate screens with one or two columns or maybe a simple table, but is ultimately very limited. That is by design because developing a UI that works well consistently is not easy. The more options you incorporate, the more complicated it becomes. This is where platforms start to diverge from no-code toward low-code.

Back to Contents

FUUZ PLATFORM
Components of a Platform

The Presentation Layer, cont.

With the lower-code options, there's more flexibility and control but now you need someone who has a little more expertise working with the toolset.

Large and complex low-code solutions, like Domo, F5, Pega and others, give you the no-code point of entry, then immediately dump you into the full code environment where you'll need to understand HTML, CSS and JavaScript to create a fairly simple screen for your users. This is not ideal, as it diminishes the value of the platform by simply encouraging the creator to use the same technologies they could have used without the platform in the first place.

Fuuz has bridged the gap by providing the ability to create a fully functional UI without the traditional limitations described and without the need for advanced code. Having said this, when the time comes for some advanced functionality, the creator can flip into advanced mode to make that happen.

The Fuuz Differentiator

Fuuz is different and unique from other platforms in many ways. First and foremost, Fuuz is a platform as a service (PaaS). Only a couple of other platforms can claim this and Salesforce.com is one of them. PaaS means that we're not just selling you a piece of software (the platform) that you have to install somewhere, administer and maintain.

The service being provided to you is our team of expert engineers maintaining the platform, the servers and everything that goes along with this, which dramatically cuts down on maintenance requirements compared to even the most stable monolithic solutions on the market.

With Fuuz, you get unprecedented up time, scalability and reliability to run your business without requiring a team of people to manage the application servers.

Many things within the Fuuz platform have been automated for our customers. This is a game-changer and does not exemplify the typical platform. For example, our data-modeling tool is a full DIY self-service drag-and-drop visual designer that allows you to create a highly complex data structure with all the bells and whistles in just minutes. This is a task that might take a developer with traditional tools days, if not weeks, of work. Not only is the time reduced significantly (literally minutes, or hours) but when you're done with the model, the platform auto-generates all of the necessary APIs for you, as well as policy level controls for all read/write functions.

Back to Contents

FUUZ PLATFORM
Components of a Platform

The Fuuz Differentiator, cont.

In most platforms the user is responsible for figuring out how to design and develop the security for the things they create. With Fuuz, the hard work is done. All you need to do is give your policy a name and select the items you want to control. This feature again saves weeks of work compared to the traditional approach.

When it comes to the presentation layer, because the APIs are auto-generated, you're building the UI within minutes of completing the data model in Fuuz. Everything is point-and-click and drag-and-drop with the screen designer in Fuuz, so within minutes you have a fully functional application.

The beauty of our business layer, or Data Flow Designer as we call it, is that there are hundreds of pre-configured business logic models for you to work with. There's even one that will create the presentation layer for you with a single click once you've made your data model.

There is no other system that has this capability. We just keep taking things a step further than any other solution because we want our customers to be successful and not limited like they have been in the past with other tools they've tried.

Back to Contents

FUUZ PLATFORM

The Fuuz Stack

Fuuz is the shining example of what modern technology can do for companies that want to grow. The Fuuz Platform has evolved like many software applications over the past several years; the architecture is nimble and flexible, allowing our engineers to take advantage of the rapidly changing technology world. As new advances in technology become available for general consumption, our team works diligently to incorporate those into the platform, enabling our customers to always be on the leading edge and take advantage of all the benefits available.

What Is a Stack?

A software stack describes the different technologies that are used for the software to be able to deliver various functionality to the end users. You will find by doing light research that most commercial software applications do not get into details about their technology stack. This is usually because these solutions have been around for a long time and there's some hesitancy to fully disclose the age of the solution and the tools.

You might see software built on Microsoft stacks; that refers to technology like MS SQL for the database, or perhaps C+ or C# code, .net or ASP. It is also common to see many older solutions like Boomi and Tibco to be built on Java. Java is a very powerful toolset but, as you well know, fraught with challenges and often the target of malicious cyberattacks.

As you research leaders in technology, you'll see other things discussed like Node.js, GO, Mongo, JavaScript, TypeScript, RabbitMQ and others. It is important to research these technologies so you can fully understand the benefits. You'll also learn that the concept of mesh architecture and micro services is extremely important for businesses that want to facilitate future growth and the adoption of tools.

Looking back to the software life cycle of most companies, when you align your business with technology that is inflexible, your business becomes inflexible. The only way around this is to start working around or outside your systems and technology, which many of our customers have done in the past. When the technology ultimately becomes stale, it becomes difficult to regain adoption.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

What Is a Stack?, cont.

With the labor challenges supply chains are facing currently, our customers are aware they need to present technology that is progressive. Younger generations are not comfortable with green-screen technology; they are a very forward-thinking group and we need businesses to adopt forward thinking technology.

What Is Fuuz Built On?

We wouldn't have done all this explaining if we were planning to tell you that Fuuz is built on 30-year-old technology but we've made the screens look pretty. On the contrary, the Fuuz Platform is running on today's technology, every day.

You read that correctly, Fuuz is designed as a mesh service-oriented architecture. Everything in the platform is real time event-based processing. This means everything can be used to automate your business, from the user interactions to integrations and even your data at rest.

Fuuz is a real cloud-based multi-enterprise application. Our customers are all running the same version of Fuuz, all the time. Fuuz is always up to date thanks to our continuous deployment model, which continuously ensures the software is constantly improved.

Other platforms installed on businesses' servers (or in the cloud) force the customer to perform updates. Have you ever been to a user group and wondered what version of the application others are using? That's one of the key benefits to Fuuz being offered as a service. It's a differentiator of our platform in the market.

The Fuuz Data Layer

The engineers at Fuuz have painstakingly selected the best tools for every job, with a focus on what our customers need today and will likely want to do in the future. We have combined the technologies that address growing needs best. While our tech stack doesn't fit the most common molds, we have evaluated the best and worst solutions to determine the sweet spot so we can deliver consistently to our customers.

You'll find that our data layer is a document-style database. We use Mongo because it offers unparalleled speed and flexibility and, when incorporated into the Fuuz Platform, allows our customers to quickly and painlessly add custom fields and records to their own apps or to our off-the-shelf solutions without weeks of development effort.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

The Fuuz Data Layer, cont.

Asynchronous processes are easily supported in this structure allowing for lightning fast user interfaces and data query retrieval for your reporting needs. Many features are supported out of the box with Mongo, such as sharding, which keeps things running fast and always presents the most relevant data. With other technology, like traditional SQL-style solutions, you'll find yourself reaching out to your software vendor to help you optimize things over time, which works but, of course, comes with the cost of end-user frustration. And your CFO probably doesn't appreciate the bill received at the end of the month.

With other databases, you need to consider execution plans, data structure optimization, re-indexing and adding more horsepower to access your data. Monolithic applications will try to mitigate this by throttling your ability to access data via various time-out limitations.

This means you will struggle to pull data or perform large-scale integrations over time. Monolithic integrations, as I pointed out earlier, often bury business logic in the API layer, which makes integrations extra painful. Without direct access to the data layer via a proper API, you will spend much more time than necessary getting things to work the way you need them to.

The Fuuz API Layer

In the world of platforms, APIs are everything. They provide the backbone to flexibility and extensibility. That's why we worked hard to select the best technologies when developing the only multi-enterprise platform for industrial companies.

GraphiQL is the Fuuz Platform's primary interface for API framework. It is a very popular solution because it's easy to use and offers robust built-in documentation. Within minutes, your team will have the ability to not only query data from Fuuz but author your own APIs that can be executed within Fuuz or from remote services.

This is unique when compared to other solutions in the marketplace because you, the end user has control. Other Software-as-a-Service (SaaS) providers are not platforms and are unable to give the consumer access to tools like this.

What you'll do instead is go through a process of authoring, or having a third-party author an API. Then, you'll need to go through various steps before you can utilize the API. At the end of the process, you still may not have access to real-time data. With Fuuz, you can author your own API in minutes and keep your project moving along, saving you valuable time and cutting cost.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

The Fuuz API Layer, cont.

These capabilities are served up by our server infrastructure built on Node.js, which allows us to create many containers or instances of our software automatically as demand increases. This is an inherent part of the Fuuz Platform. It's what our solution provides and it's part of the service you're offered. Traditionally, as businesses grow and system usage increases, companies would have a team of people focused on optimizing throughputs, increasing storage size, CPU and memory utilization to address the organization's growing needs. These types of things go away with a platform like Fuuz.

We also leverage RabbitMQ as a message-brokering service to provide event-driven architecture so that your apps and solutions are real time, unlike many other solutions you may find on the market.

Traditionally, if you're viewing a form like a customer record in your current application, what happens if another user were to manipulate that data without you knowing it and then you make an edit? A data collision. The Fuuz Platform's architecture enables robust APIs and event systems. We can avoid data collisions and even provide indicators to users alerting them when someone else is viewing the data or if the data has been modified. These features are automatically built in so there's nothing you need to do to incorporate this. Our solution saves days of development effort for every screen you may want to build. K8 or Kubernetes orchestrates the creation and removal of all these magical containers that are working for you behind the scenes.

Our multi-enterprise infrastructure means your system is not operating on an island, or your own server; we're leveraging real cloud power across hundreds of customers. You get nearly unlimited potential from a computing standpoint. With other platforms, you may be responsible for managing this and will end up paying a bill from AWS or Azure at the end of your month, not to mention the cost of internal IT needs.

The separation of these critical components, from an architectural perspective, also makes the Fuuz Platform extremely easy to maintain and scale. No major upgrades are ever required. In fact, the only upgrades you'll ever see are routine performance enhancements and new features or modules that our customers can choose to take advantage of, or not. Fuuz has modern, lightning-fast technology under the hood that is inherently better for our customers' experience. Queries take a fraction of the time and return 100x more data than you can get from antiquated software using SOAP-based APIs.

All of the APIs in Fuuz are REST; everything is streamlined into JSON objects, which is most popular nowadays. If you're interested in introducing technology that will be relevant to the incoming workforce, there's nothing better than Fuuz.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

The Fuuz API Layer, cont.

Why implement something that's not taught anymore? With Fuuz, that's not a concern and never will be. Technology is in perpetual flux and we incorporate new technologies into our stack continually. For example, as scripting languages evolve from JavaScript to JSONata to Python, they become available for you to take advantage of in the platform.

Fuuz will never be a legacy technology. The advantage here is that while we maintain modern approaches to everything, the legacy methods never go away, so you get the best of both worlds. If there's no reason to change, you don't have to!

The Fuuz Business Layer

The business layer is where the magic happens, right? The data layer is somewhat exciting, the API layer topic maybe not so much, but the business layer is what our customers are all about. The business logic layer of any application is the heart and soul. It's literally the "why" of "how" your company operates.

The approaches discussed so far are the full-customization side of the Fuuz Platform. Configuration is different. In the configuration approach to software, you are provided binary choices; you either opt into something or not, and that drives how you use the software. This, of course, has its pros and cons.

The most notable positive is that you're going to get a solution up and running relatively fast compared to something like SAP, Oracle or MS Dynamics, where everything takes some design and development effort. The downside to a configuration-only product is, of course, that you have to accept the fact that you'll need to operate your business like every other company using that software product. That may be fine for some companies but, as we discussed in the software life cycle section, this doesn't work for everyone.

The Fuuz engineering and delivery team has taken a completely radical approach to the configuration-only dilemma. Fundamentally, the platform includes what we call our Data Flow Designer. It is an orchestration tool that allows users to build business logic for anything, including integrations, documents, business processes and user interaction with drag-and-drop tools. Some knowledge of coding languages is required, but there are multiple scripting languages you can leverage such as JSONata, JavaScript, TypeScript and Python, with more on the way as new options become popular and widely supported.

That said, Fuuz Platform users do not need to write every line of code. That's because our engineers have abstracted the most common use cases and made them into what we call "nodes." Nodes are common practices for manipulating data and processes that you can drag onto your canvas and simply wire into the flow.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

The Fuuz Business Layer, cont.

This saves you hours, days, or even maybe weeks, of effort compared to other tools.

Our team has also developed thousands of business flows for repetitive actions such as integrations with other systems like Plex or SAP. In addition, there are predefined application flows for common business logic. That's why Fuuz applications can handle things like inventory depletion and genealogy, printing documents, posting production, moving inventory, and the list goes on.

These become building blocks for our delivery team as well as our customers, so there is little need for reinvention. These flows were developed by Fuuz, eliminating the need for in-house knowledge and skills that some of our customers may not have. The concept of reusable code components is a significant differentiator for the Fuuz Platform. That's because it accelerates application development time, saving both time and money for our customers.

We've discussed configuration versus customization previously and how Fuuz takes customization to a level never seen in other software platforms. Since our tool set is highly flexible, most of our customers and our delivery team incorporate configuration functionality into standard apps. With MES, for example, our customers can make binary choices such as whether they want to log multiple operators into a station. Every Fuuz app has hundreds of similar choices.

With Fuuz, configuration is not buried in business logic like it is with monolithic applications. It's part of our data layer, API and is ultimately incorporated into the business logic or flow as a simple variable. Configuration options are easily managed by authorized users and don't require coding; just change the option and update.

The best part is that Fuuz customers can add other configurations to the solution quickly and easily. But it's so simple to tweak processes and manage changes with the Fuuz platform Data Flow Designer tool, external configurations are rarely needed.

Let's recap. There are pros and cons to configuration only and customization only. Our team has evaluated all of these and we kept all the pros alive in the Fuuz Platform. This means you have the simplicity of configuration, but the flexibility of full customization if you need it.

You've probably heard that customizing software leaves you on an island or makes upgrades and updates difficult. That's generally true but not with Fuuz. That's because the Fuuz Platform is continuously deployed and updated with no impact to the end user – unless there are new features you want to incorporate.

Since Fuuz is a multi-enterprise platform, all of our customers are running the same version or release every day.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

The Fuuz Business Layer, cont.

The real magic of Fuuz is how we have isolated the actual software or platform from your apps – your MES, WMS, TMS or your custom one-off solutions you've built using the platform. Your app-specific customizations are never impacted by our continuous approach to enhancements.

Azure and AWS also continually deploy changes, and you can build your own solutions within them. But their changes could, and likely would, impact your applications because they change authentication methods, network strategies or the migration process.

Since Fuuz is a PaaS that manages the back end completely, our customers have complete flexibility and control over how their users use the apps. And our customers never end up with siloed software living on a separate island.

The Fuuz Presentation Layer

Last, but certainly not least, is the Fuuz Platform presentation layer. If you're shopping for software, this layer is probably the focus of your evaluation. What you see in a software demo is usually what you get when you purchase that solution.

That is not entirely true with the Fuuz Platform. A lot of what you see in our demos is what you'll get with our out-of-the-box applications, but Fuuz also provides the layers of personalization or customization that you need to be truly successful every time.

Depending on your subscription package, you may have access to our Screen Designer tools. This set of tools is a drag-and-drop design application that allows you to build amazing UIs for your business in minutes. All you have to do is tie them to flows that you've created or pulled from libraries and you're off to the races.

Let's say you need to add more data to a product record. Go to the Schema Designer, add the columns and save it. Then go to the Screen Designer, pull up the screen those fields need to be added to, drop them in place, set your defaults if needed, save it, deploy the latest version and let your team know they can start using those fields. All of that likely took place in under 5 minutes.

Can you do that with a monolithic application? Unfortunately, no. If the software vendor – and that's a big if – agrees to add some fields for you, then you'll likely wait months or years and be asked to participate in some co-innovation project where you'll be paying them mightily for adding the fields. That may not bother everyone, but now that you have the fields, what are you going to use them for? Hopefully you mentioned to the vendor that you'd like to use them in an API, otherwise that's a new project. Maybe you needed them to trigger some activity on an operator dashboard, likely another project as well. Maybe two years later, you have those three additional fields that were so important to your operation.

Back to Contents

FUUZ PLATFORM
The Fuuz Stack

The Fuuz Presentation Layer, cont.

With Fuuz, you added the fields in 5 minutes. You adjusted business logic in 15 minutes and you had them on your operators dashboard before lunch that same day.

This storyline really is about the underlying technology. The Fuuz front end, as our engineers call it, uses React and Redux technology. You may be familiar with this, as you've likely heard Facebook, LinkedIn and several other highly visible cloud apps use React. React is highly flexible, scalable and fast.

One reason we like React is that it enables lightning-fast user experiences. React installs a very small file into the web browser when a user initially visits a site. This file contains a lot of cool stuff – mostly framework-type things that can run locally on the client instead of having to go back and forth over the internet. This significantly reduces latency and bandwidth on your network.

If you've ever wondered how Facebook is able to automatically start and play videos while you're scrolling through without the typical jerkiness, there you go. The experience most of our customers have with business applications like ERPs, WMS is quite different. Customers often wait minutes for screens or reports to render; there's a lot of activity going on between where you're sitting and where your data is. We can change that, and have, using modern technology.

Many of our customers don't have to do a lot of activity within the Fuuz Platform's screens because we also extend our existing applications. This is literally a Fuuz-only capability; any application you have that's accessible via a web browser can be augmented by Fuuz using our Browser Extension tool. Your users never have to login to multiple applications, have a bunch of apps open, or toggle between tabs.

Our Browser Extension can also leverage data that's available on screen from these existing applications, essentially turning some stateless, monolithic solutions into real-time powerhouses.

This enables your team to work much faster and with much more data available at their fingertips. While you can't add fields or manipulate data models in most monolithic apps, you can augment those apps and add whatever fields you need with Fuuz. You can also use the Fuuz Data Flow Designer and Screen Designer to go even further. If you love your current application but you're not in love with its lack of flexibility, Fuuz can fix that.

Back to Contents

FUUZ PLATFORM

The Fuuz Architecture

The Fuuz Platform is a multi-tenant, multi-enterprise PaaS that supports multiple environments. That statement is worth unpacking because this is how we keep our solution running at lightning speed, regardless of your computing intensity, and it's how we keep our customers' assets safe.

Architecture Basics

There is nothing unique about being a "cloud-based" application anymore. Any software company that realizes the benefits cloud has to offer is moving in this direction or has already. While Fuuz is cloud-based, this is not something we feel is noteworthy anymore, just like talking about how long the application has been running on the cloud. That's become irrelevant because it's a dated concept.

There are, of course, various degrees of cloud-based applications. Many are simply hosted in the cloud and running on AWS or Azure. While this provides some flexibility you may not have on-premise, you're ultimately just running your application on another piece of hardware that you don't own. You still have updates and hardware upgrades to consider, as well as point-of-use costs. In a hosted environment, you're not getting a service. You're just renting the hardware from a service provider, so you only get what you're willing to pay for.

The concept of multi-tenancy is something that was made up decades ago by software companies that were some of the first to move to the cloud concept. The idea behind multi-tenancy, in very simple terms, is basically one giant server that supports many customers or users.

The advantages favor of the software vendor. That's because the costs are shared across many customer accounts. Some customers utilize the software heavily while others do so lightly. It's easy to think that the usage averages out and becomes a win-win. But the heavy users aren't necessarily paying for what they're using and the light users get the benefit of paying less than they would if the application were hosted on a standalone server.

To further complicate the situation, some software vendors started to come up with creative ways to reduce costs further. One way they do this is with multi-tenancy databases that store every customer's sensitive information in the same place.

Back to Contents

FUUZ PLATFORM
The Fuuz Architecture

Architecture Basics, cont.

This limits the flexibility of the application because every customer is forced to use the same data structure. Not only does the application become less flexible, but such databases also create a security risk for customers. All hackers have to do is crack the code once and they have access to everything.

The Fuuz Platform was designed with advanced security in mind. Not just the cyberattack kind of security but also browser security. Companies like Google continue to make their Chrome project more and more secure. That's why companies like Microsoft gave up and are using the Chromium project as the core for their browser now. That said, one of the most significant risks is malicious access to your PC, network or other infrastructure from a browser application.

Your users typically have a lot of freedom to access various websites throughout the workday. At lunchtime, perhaps they jump on YouTube or Facebook. These employee activities are not malicious, but someone else out there is trying to gain access to everything that the employee's computer has access to.

Browsers continue to improve and become more difficult to penetrate. But this process presents a problem for cloud-based solutions when it comes time for tasks like printing paperwork or barcode labels in manufacturing environments. They simply cannot do it.

This creates additional work and slow-down issues on the shop floor. Some cloud platforms have worked around this by requiring that their customers use certain operating systems that support their plug-ins. And this can become a significant hidden cost when deploying an application. Others providers will suggest the use of a third-party application to bridge the gap. And that becomes an advanced science project to get the integrations to work with your processes.

To recap, what is the difference between an on-premise application and one that's hosted? Nothing. It's the same thing unless you're renting equipment. All the concerns and issues remain the same. There are risks in traditional approaches to multi-tenancy from a data security perspective. Ultimately, we suggest you do your homework on what you're buying. It may look wonderful on the surface, but what lurks under the surface may cause your business significant trouble down the road.

Multi-Enterprise

The Fuuz Platform is a multi-enterprise solution. This means that each of our customers is safe and secure on their own isolated data farm. Every customer's enterprise is completely isolated from a data standpoint, ensuring the utmost in safety and security for their data. This also provides an unparalleled ability to create and maintain unique data fields, and the separation between customers ensures that noisy neighbors do not become a problem.

Back to Contents

FUUZ PLATFORM
The Fuuz Architecture

Multi-Enterprise, cont.

What do we mean by noisy neighbors? Think about the multi-tenant model. Such an environment includes companies of all sizes, all of which are involved in transaction volumes of varying intensity. Users don't get to choose how their systems are provisioned and may end up next to another company with ultra-high transactional volumes that impact their neighbors' user experience from a performance perspective.

This is not fair, but it's one of the downsides to a multi-tenant approach. Unfortunately, companies usually don't realize there's a problem until the solution goes live and, at that point, the vendor can't do anything about it. We hear this all the time from customers switching to Fuuz from other vendors.

Having your own island of data is a blessing. We've said Fuuz is everything a cloud solution should be and nothing it shouldn't be. With your own island of data, everything becomes blazing fast and your experience is never compromised by other customer's usage of the system. The thought of someone else on the system, potentially gaining access to your data, is also never a concern.

Tenants

The concept of our multi-enterprise solution is that each customer is in control of their destiny with the platform. Your subscription allows you to consume as many resources as necessary to run your operations.

Within an enterprise, we have what we call "tenants." Think of this like a giant apartment building. You own the building and tenants reside in the building and share certain amenities, like the parking garage. Fuuz is designed so that every tenant has isolation within the building. It's a lot like soundproof insulation in the walls between each of the apartments in your building. You may have a rock band practicing next door, but you'll never hear them and can continue binge-watching your favorite Netflix show without interruption.

In the same way we protect our customers from other customers on the platform, we also protect our customers from themselves. You may have a need for different configurations or customizations for different facilities (tenants). What if you have mixed-mode manufacturing going on in the same facility? Maybe your organization is doing discrete and continuous manufacturing under one roof. Not all applications are up to the task.

With a monolithic application, users only get one configuration for each location. The only solution is to purchase, implement and configure multiple environments of the application to accommodate different processes, which is time-consuming, frustrating and leads to disparate systems within the same facility.

The Fuuz Platform is built around the concept of flexibility. You can keep everyone on the same page (one source of truth), all within the same tenant.

Back to Contents

FUUZ PLATFORM

Environments

If you've ever done development or have worked on a project requiring development, you have likely run into the situation of multiple environments. Fuuz is no exception to this. In fact, most of our customers have at least two environments, if not more, depending on the subscription package.

Build Environments

Build environments are exactly what they sound like. This is where you can install pre-built apps, extend those apps or even start your own apps from scratch. Users can connect to other software applications using our Flow Designer and iPaaS connections, design data models in the Schema Designer and then, of course, edit their UI in the Screen Designer. This provides a safe (disconnected from production) environment where you can experiment until you have things the way you want them. The process never interferes with your users or production system.

The Fuuz Platform's build environments last as long as you want them to, which enables our customers to work on prolonged projects. Other systems may offer these types of environments, but they refresh every 24 hours and give users very little time to complete the work.

We also apply platform updates to build environments first. They regularly receive enhancements to improve performance and introduce new features. Our customers have the perfect playground for evaluating new features before making them available to their business users.

Quality Assurance Environments

Quality assurance (QA) environments are not included with every subscription package but, for those who have them, they provide a seamless user acceptance testing (UAT) environment and give our customers the ability to maintain a constant environment between build and production.

You can deploy changes in QA environments for your business users to test, validate or even train new employees without impacting production. This also eliminates the need to constantly rollback changes on your build. Very few software platforms offer this level of control.

Production Environments

Production environments are where user transactions, live integrations and other daily activities live. Most transactional volume happens here. You can replicate your production environment to your QA or Build upon request – and it doesn't take nearly as long as what you may be used to. We can replicate the environment in one day or less; you won't have to wait three months for a refresh.

Back to Contents

FUUZ PLATFORM
The Fuuz Gateway

The Fuuz Gateway is another piece of the architecture puzzle. While some applications and browser security requirements require users to stick with specific operating systems or hardware, Fuuz does not. We've taken a modern, reliable and scalable approach to these concerns by developing the Fuuz Gateway application. It's a little bit like AWS Greengrass.

The Fuuz Gateway has a couple of deployment options:

1. At the Edge

You may have heard this term before. It simply means something that is deployed as close to the point of collection as possible. In the world of manufacturing, the "edge" is usually the piece of equipment or workstation where data is being collected. By deploying at the edge, you can collect more data and do so more reliably than you can with other applications that use centralized deployment.

At the edge commonly refers to high-volume machine data-collection activity. One advantage of this method is that the user is not dependent on a single point of failure. It also reduces the overall cost of ownership, as the hardware required at the edge is very inexpensive compared to servers and virtual machines that require complex hardware, software and expertise to maintain.

2. Centralized

Centralized is the most common approach of deployment for solutions that interact with manufacturing equipment. Kepware, Wonderware, Ignition, FactoryTalk and ThingWorx all use a centralized-deployment model. These types of software run on a physical or virtual server somewhere in the building that is connected to two networks; one is an industrial network and the other the business network, which usually has internet access. This model is often used for low-volume machine data collection, printing applications and disk-level integrations.

There are pros and cons to both deployment models and each needs to be evaluated to ensure that all requirements are met. One advantage of the Fuuz Platform license model is that edge deployment does not incur an additional cost. The Fuuz Gateway can be installed on almost any type of device, as well, so the cost to deploy hardware is drastically reduced.

The Fuuz Gateway provides a secure tunnel between assets behind our customers' firewalls and the Fuuz Platform. Pub/sub technology keeps everything in sync in real time. Also, if the project involves any type of data collection or aggregation, the Fuuz Gateway has built-in store and forward features to circumvent latency and network issues. Robust logging allows users to analyze any inconsistencies in their integrations.

Back to Contents

FUUZ PLATFORM
The Fuuz Gateway

One of the best features of the Fuuz Gateway is the fact that once it is installed locally, it never has to be touched again aside from updates. Unlike other OPC UA, printing solutions and disk-monitoring applications, the Fuuz Gateway is always in sync with your Fuuz Platform environments.

The platform's cloud interface makes it easy to add and configure devices, view logs and see connectivity data. The process eliminates the need for VPN and RDP access, making projects much easier to deploy and enabling better support. If you need to add more PLC tags to the scope of the store and forward system, for example, you do that from the cloud by checking a box and clicking "update." The Fuuz Gateway is updated instantly and starts to store values should the user experience a network issue. Every other solution that offers edge-capable software requires much more work to make changes.

FUUZ PLATFORM
The Fuuz Browser Extension

The Fuuz Platform's Browser Extension feature is an industry first. This little application is available to be installed on any browser and brings the functionality of the Fuuz Platform into any browser-based application.

Unlike other solutions that depend on batch processing, you can use any feature of the Fuuz Platform and extend it into your existing solutions via the Fuuz Browser Extension. This eliminates complex training, as well as the need for users to log into multiple apps or work with multiple tabs.

The Browser Extension provides the ability to augment and extend existing applications with screens and processes built in the Fuuz Platform. It also allows users to perform integrations with outdated software applications in real time. The Browser Extension can read anything on the application in the browser and pass the information back to the Fuuz Platform to trigger events.

Customers can also leverage the extension to augment information in monolithic apps. Our solution gives users creative ways to add custom fields or records to existing non-flexible solutions, which extends the value of their existing technologies.

Fuuz is the only xPaaS developed by manufacturing experts to solve the complex problems of industry, and now you know a little more about how we do just that. For more information about how the Fuuz Platform can support your business, please call MFGx at 1-877-801-FUUZ (3889) or contact sales@mfgx.com.