digitalML
Software for the New Enterprise

How The ignite Service Design Platform Supports Your API Lifecycle

ignite | Service Design Platform

Mike Gilpin, Chief Customer Officer
DigitalML— 580 California Street, Suite 1600, San Francisco, CA 94104
US: 415-373-0302 **UK:** (+44) 1344-390542 **E:** jbogard@digitalml.com

Executive Summary

Feeling under extreme pressure to accelerate delivery of your API portfolio? Blame Jeff Bezos. He is widely viewed as having kicked off the race to build the API Economy with his mandate to Amazon developers in 2002 that:

- All teams will henceforth expose their data and functionality through service interfaces.
- Teams must communicate with each other through these interfaces.
- There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
- It doesn't matter what technology they use. HTTP, CORBA, Pub/Sub, custom protocols – doesn't matter. Bezos doesn't care.
- All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
- Anyone who doesn't do this will be fired.

Today we live in a world in which most architects and technology business leaders take for granted that they are competing in the API Economy for the future success of their Digital Business, driving rapid growth in APIs (see Figure 1).

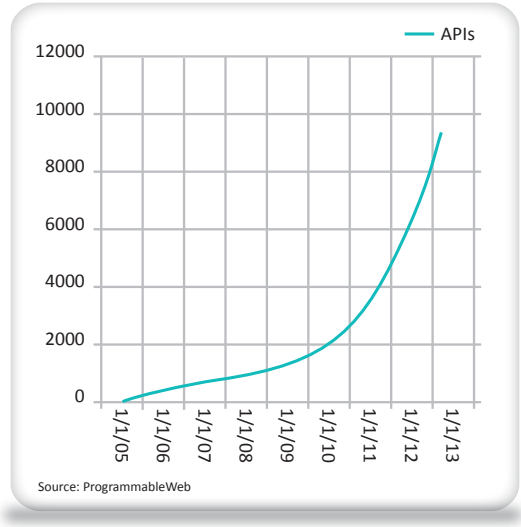


Figure 1: APIs Are Experiencing Exponential Growth

Amazon was building its services on a green field, but large enterprises often struggle to deliver APIs while stuck in the concrete of significant technical debt. How do leading large enterprises today manage to succeed despite these obstacles? They use the ignite Service Design Platform to help transform their organization to use a business-led approach based around a *model of canonical data and services*. This speeds delivery of the most strategic parts of their API and service portfolio, while providing effective governance to help ensure the quality, understandability, and reusability of those APIs and services. Read on to learn how ignite uniquely supports and melds with your API design and delivery lifecycle, to guide your strategy in maximizing the value from using ignite as part of your overall API solution.

Contents

ignite Speeds The Service Design Lifecycle

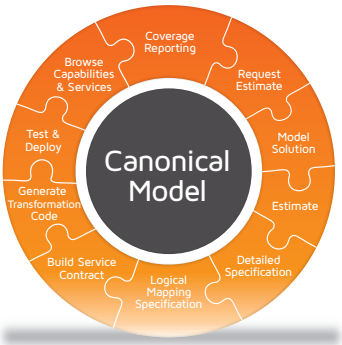
The ignite Service Design Platform automates the oft-neglected *design* portion of the API and service lifecycle, but also can seamlessly integrate via its own APIs and product capabilities with the rest of the API lifecycle (see Figure 2). Successfully exploiting the full capabilities of the ignite product suite, together with all the integration opportunities it affords, will equip your firm to win the API race, supporting a fast-moving business strategy.

Firms that win the API race make API design and creation a core competency, requiring a new approach to managing the API and service design lifecycle. Decades of siloed investments in IT coupled with multiple layers of newer web and mobile architectures and middleware make it uniquely challenging for large firms to win the race, due to a chaotic API, service, and data portfolio. How to cope? Leaders in breaking this logjam have succeeded by creating a business-led portfolio of APIs and services that consistently deliver their desired business outcomes. ignite uniquely supports this effort, through:

- Basing the API design approach around a powerful *model of canonical data and services* that maximizes leverage.
- Empowering business owners to drive *rapid design* of the key APIs and services for their products.
- Enabling technology staff to *collaborate* over the entire service lifecycle to deliver the optimum API and service portfolio without needless duplication or silos.
- Equipping development teams with the *automation* they need to speed past design into developing and delivering the APIs and services that bring business results.

Figure 2: The API & Service Lifecycle As Addressed By ignite

ignite's Secret Sauce:
Your Canonical Model Of Data And Services



ignite is the only suite of tools for managing your API design lifecycle and speeding API delivery that provides these capabilities on a solid foundation provided by your *model of canonical data and services*. This is the secret sauce that enables the largest enterprises in the world to gain control of API delivery using this canonical approach, uniquely combining speed and governance.

Why is governance of APIs to conform to a canonical model so important to these firms' success? On the simplest level, it's because the "R" in REST stands for Resource, which is an abstraction of *data in motion*. For a portfolio of RESTful APIs to deliver the business value enterprises demand, it must promote a coherent view of these Resources – but without the right support from tools, this is so challenging that many organizations have given up trying, letting development teams build another layer of technical debt in the form of a chaotic collection of APIs.

In contrast, firms using ignite are able to strike a balance between allowing the necessary degree of diversity APIs require while aligning APIs with a coherent model of reusable resources – data – that the business can understand, and that customers can rely upon to support consistent experiences over many contexts. Firms' most strategic APIs – the ones that expose core capabilities, or the ones that business partners and developer ecosystems consume – require the closest alignment, but all APIs expose some canonical data, so benefit from loose coupling to the model, so that:

- Where the API exposes information that's in the canonical model, it maps to the canonical representation, naming, and semantics. Failing to do this just introduces needless complexity and confusion.
- Where the API has its own information local to its implementation, it handles that as local elements and attributes. Model managers can periodically review these to determine whether to evolve the canonical, but if the data is specific to the local context, may well decide to leave it local to the API or application.
- As the API portfolio evolves, IT gains powerful visibility into *data lineage* and the *impact of change* across data sources, services, and API operations (as captured in ignite's repository). It's only through capturing this information as a natural side-effect of developing and delivering APIs and the apps that use them, that digital firms of any size can hope to cost-effectively sustain their digital business.
- Developers can use their understanding of the data they require to find the APIs they want to use. ignite makes it easy to search for APIs based on any information the developer knows about their requirements, from business names of information, to names of other applications or services that use them or are called by them, to text in requirements documentation including names of other systems or databases. Powerful searching is one of the biggest payoffs of using ignite to manage and automate your API lifecycle.

digitalML has delivered all this power in a form designed to integrate with the rest of your API lifecycle tools and environment, providing its own APIs that expose the rich contents of ignite's repository for your use in many ways that digitalML could never anticipate, enabling you to mold ignite into what *you* need it to be. Read on to learn how you can not only leverage the power of your canonical model in ignite, but also connect it to the rest of your API world.

ignite Enables You To Manage Your API Portfolio More Effectively

Effective API and Service Portfolio Management begins with the right strategic framework based on your required *Business Capabilities*, combined with effective integration with API Portals and your Project Management environment (see Figure 3):

- **Business architects define Business Capabilities.** ignite enables you to use our API to import the Business Capabilities your business architects have defined. Each capability exists within a hierarchy, from high-level to low-level capabilities, with associated descriptions and metadata. Firms begin their setup of ignite by importing their Business Capability catalog, so that as new services are defined, analysts map them to the Business Capabilities they support. This makes services easier to find and reuse, and redundancy easier to avoid.

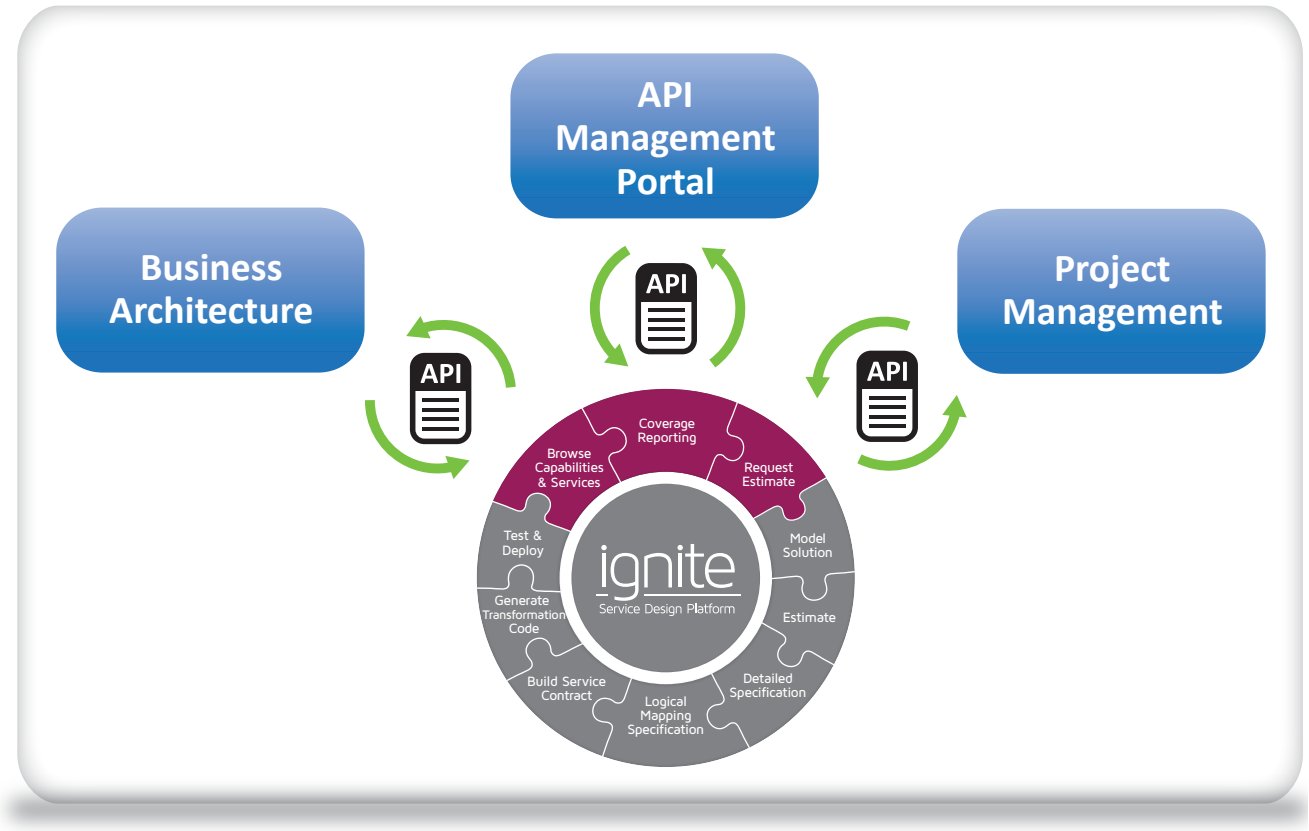


Figure 3: ignite Becomes The Core Of Managing How APIs & Services Deliver Your Business Capabilities

- **ignite integrates with your API Management Portal.** Some firms have used the ignite API to establish a real-time bi-directional link between ignite and their API Developer Portal. Uses for such portals range from publishing API documentation to developers to ease their API usage, to enabling business stakeholders to explore the API portfolio, to feeding developer tools such as Swagger sandbox (as already provided within ignite). digitalML has already built a sample implementation of two-way integration with the API Management tool WSO[2].

- **ignite can integrate with your Project Management tool.** As part of the service planning process, ignite enables architects and analysts to create Solution Models, which map to projects or programs, and to capture other planning data such as estimates. By using the estimating capabilities within ignite, this information can be integrated with your Project Management (or Portfolio Management) tool, using the ignite API.

- **ignite ServiceStore makes this management information, and more, readily available.** ServiceStore enables managers to track the progress of design and implementation of your API and Service portfolio by Business Capability or other management views, centered on the overall portfolio rather than just projects (see Figure 4). These views include tracking the state of individual services and operations, as they move through the lifecycle. ServiceStore also keeps track of how long each service or operation took to move through each stage of the lifecycle, enabling managers to spot patterns in any bottlenecks, or to rebalance resources to speed more critical services to market.

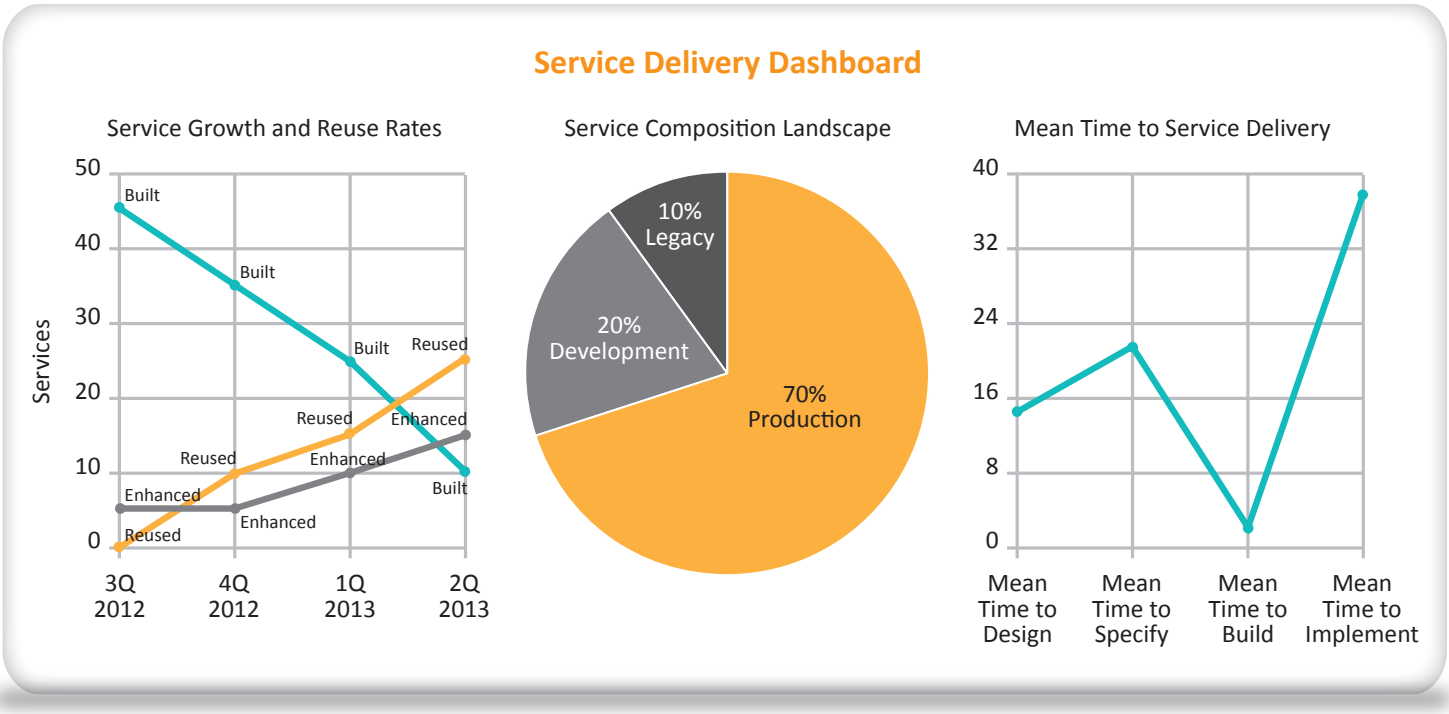


Figure 4: ignite ServiceStore Delivers Powerful Management Dashboard Views

Bottom Line: ignite offers a broad management perspective of the API Design Lifecycle not available from other tools, and also plays an important role in integrating information across other parts of the service lifecycle managed by other tools, enabling for the first time an end-to-end view across the service lifecycle for the entire API and service portfolio. It is only with such a comprehensive view that you can ensure that your precious resources are correctly allocated to deliver the most important services first, while also bearing in mind project- and program-level priorities.

ignite Accelerates Your Teams Through Collaborative API & Service Design

ignite integrates with the tools and artifacts your firm uses to manage the design work analysts and architects are doing in ignite, via its API and related capabilities (see Figure 5). This integration increases collaboration among managers, analysts, architects, and developers, enabling you to achieve good governance and design practices more efficiently and effectively. It also significantly decreases the cost and effort of producing, modifying, consuming, and managing API documentation, streamlining the whole process of producing and consuming APIs and services by you and your partners.

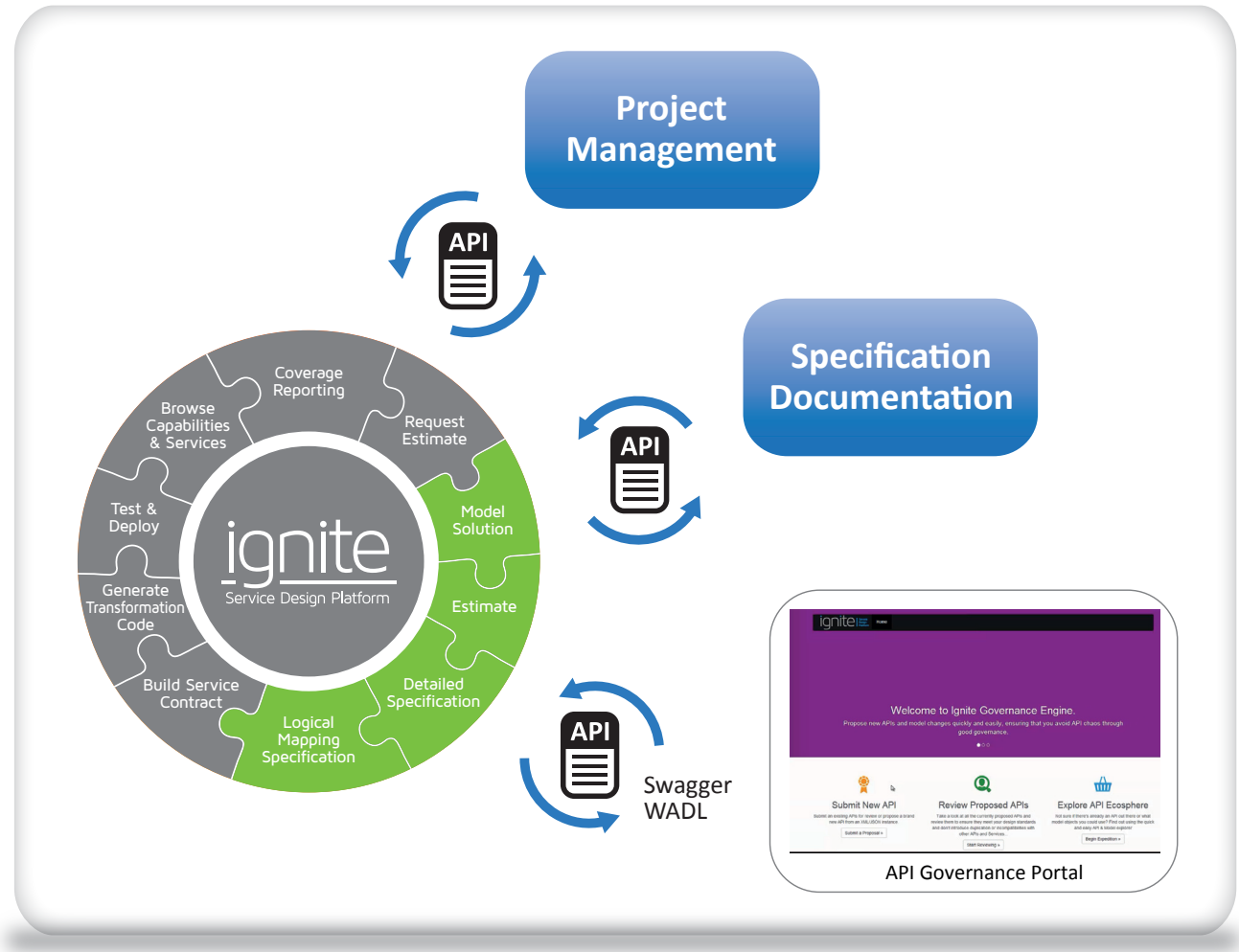


Figure 5: ignite Provides A Solid Foundation For API Governance, Maximizing Impact Of Investments

Based on what digitalML has seen our customers achieve using our API and other integration capabilities, consider using this approach to:

- **Integrate more deeply with your Project Management tool.** As mentioned above, ignite Solution Models and service-planning data such as estimates can be integrated with Project Management tools. Moving into the service design phase, more such integration can, as services move through lifecycle stages, mirror progress back into Project Management, enabling project-level views to track work progress and completion for services being designed in ignite. Likewise, information about dependencies and other risk factors that naturally arise as designers use ignite can be used to increase the depth of information that project managers have about service design efforts.

- **Integrate with or even replace API documentation.** Firms can streamline specification production by creating API links from ignite into Microsoft Office or other similar tools. And as your firm moves forward with its implementation, with more and more information online, you can likely replace these documents entirely with some combination of ignite's off-the-shelf capabilities, and custom extensions built on ignite APIs.

- **Integrate with an API/Service Governance Portal.** Customers have already integrated ignite via its APIs with API governance portals. digitalML has also built a demo governance portal as a proof-of-concept, showing how a customer can use the combination to ease the review of proposed services and speed up the whole governance process. As soon as a service appears in ignite in the "proposed" state, it becomes one of a group of proposed services available for review by the Service Design Office in the portal. Reviewers access full documentation about the proposed service, the requirements it aims to satisfy, the data it will manage, and a first-cut view of its operations. If the SDO approves the service, its state in ignite updates in real-time, and the analyst can proceed right away with further design effort, achieving **speed *and* governance**.

Bottom Line: ignite can play a central role in your API design and governance process, speeding API delivery and lowering its cost. It can also help jump-start your efforts to deliver an API developer portal, as it has already done for other firms, making it easier to build a developer community, promote the use of your APIs, and ease their consumption by internal and partner developers.

ignite Generates API & Service Artifacts For Development Teams

ignite has rich capabilities for delivering API and service artifacts through its download packs, which can be consumed through the file system, or via the ignite API (see Figure 6). The list of formats ignite supports continues to grow, with support for Avro (a Big Data format commonly used with Hadoop) added in the latest release (5.6.3.0). As the list grows, the leverage you can get from its use of ignite grows as well, compounding the return on the investments you make in modeling and service development.

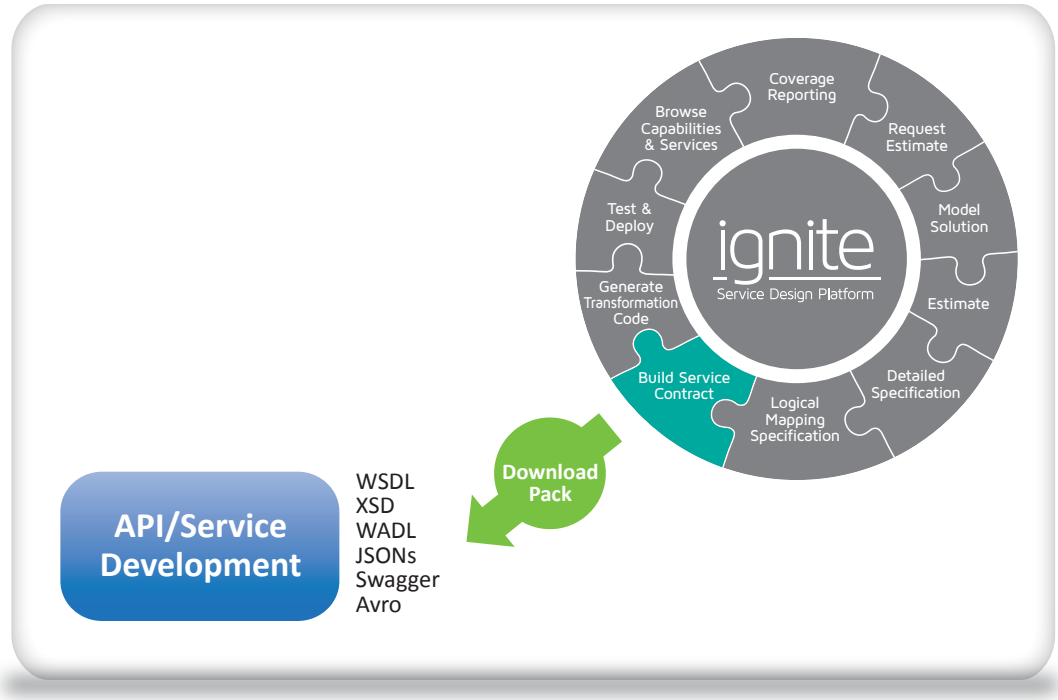


Figure 6: ignite Automates Iterative Delivery Of API/Service Artifacts Throughout The Dev Stage

- **Output Templates bring greater consistency and automation.** ignite generates artifacts governed by rules laid down in Output Templates you define in ignite. This increases the leverage of ignite's automation, while ensuring the artifacts comply with your local standards – without relying on humans to review services against those standards.

- **Using ignite is orders of magnitude faster to deliver these artifacts.** Customers regularly report that a process to get a WSDL or WADL that used to take weeks can now be completed in a few hours.

Bottom Line: API and service artifacts are where "the rubber meets the road" in development, and ignite delivers.

ignite Generates Executable Mappings For Middleware & Appliances

ignite enables analysts to define reusable mappings between canonical data definitions and the underlying physical systems that provide that data (see Figure 7). An optional part of ignite's download packs, it's worth understanding their role distinct from other artifacts because of the opportunity to use generated XSLT to reduce coding effort and further speed API and service delivery:

- **XSLT declaratively specifies powerful transformation semantics.** Although other tools may be capable of generating XSLT, none of those other tools are integrated with the powerful modeling capabilities of ignite. Using ignite to generate XSLT leverages the investments firms make in modeling, and enables the return on those investments to be leveraged more widely by reusing mappings and transformations across multiple services.

- **ignite-generated XSLTs deliver best-in-class performance.** Some other firms had given up on using XSLT as part of their runtime because of sub-par performance, but upon trying ignite-generated XSLT, found they delivered superior performance that met their design goals. This is because ignite delivers XSLT shorn of needless overhead, based on carefully trimmed data structures. Provisioning this XSLT into appliances such as IBM DataPower increases their performance still further, through the benefits of engineered systems architecture.

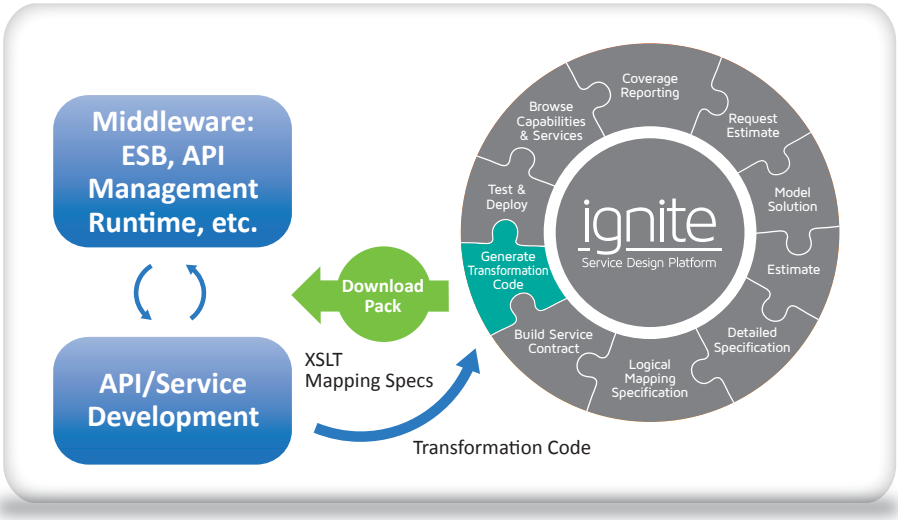


Figure 7: ignite Generates XSLT For Transformations Among Services & Legacy Systems

- **ignite allows developers to extend XSLT with ignite-managed code snippets.** Being declarative makes XSLT powerful, but some transformations require programmers to write a bit of code to overcome the limitations of a declarative approach. ignite makes the creation and management of those code snippets part of its managed approach.

Bottom Line: Firms should use ignite's support for mapping and transformation to increase automation of API behavior.

ignite Connects With The Rest Of Your API Lifecycle Environment

Although ignite is primarily focused on adding leverage to the design and build part of the API lifecycle, years of working with advanced customers has brought additional features that enable ignite to be optionally integrated with even more parts of the downstream lifecycle (see Figure 8):

- **ignite can be part of your testing solution.** Although ignite is not a testing tool, it offers support for testing in two important ways: 1) For API development, customers are using the Swagger artifacts ignite generates to drive API sandbox testing (via SwaggerUI), enabling developers to validate interface designs before coding; 2) For firms implementing Service Virtualization, ignite delivers artifacts and metadata that these tools can use to simulate services in stubs before they are built, or to present simulated workloads to services under test to verify their performance.

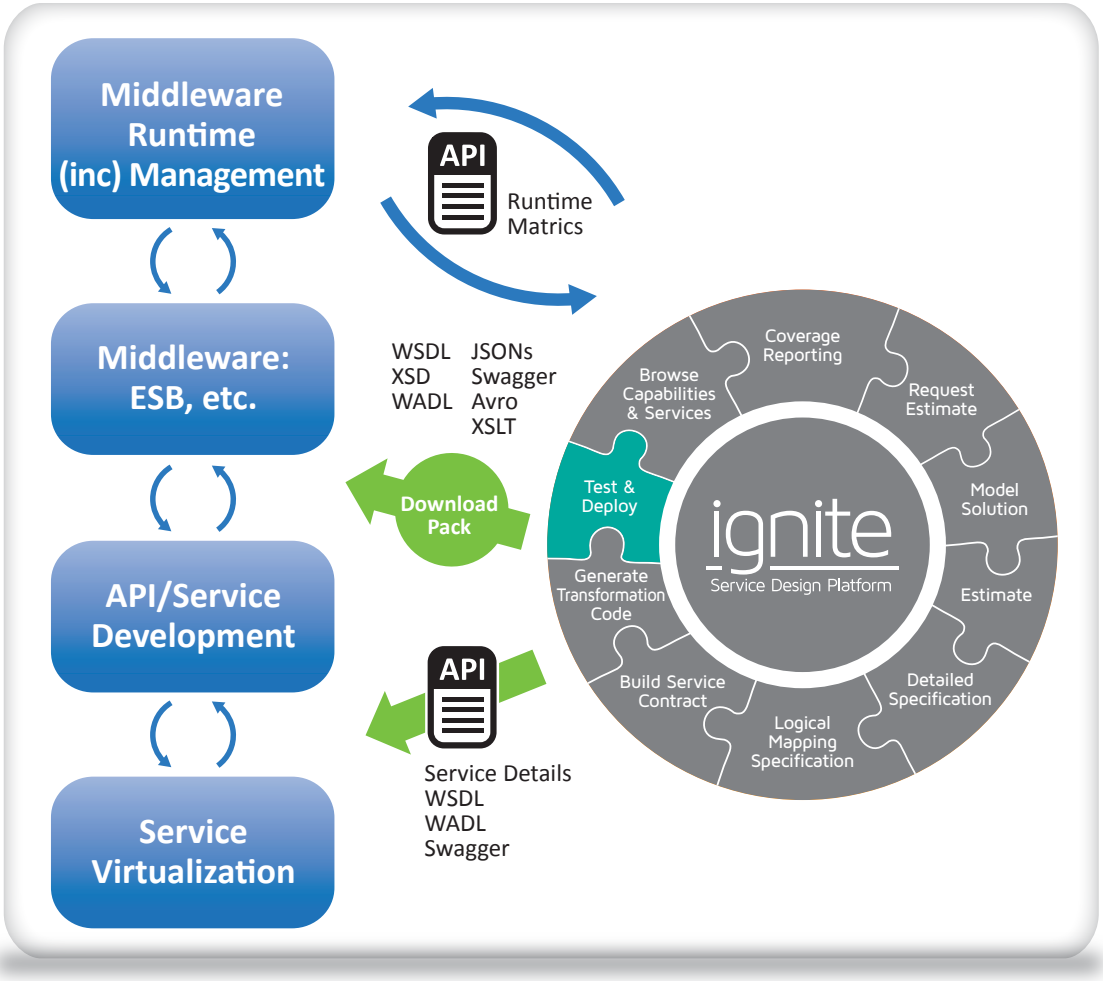


Figure 8: ignite Integrates With Runtime Environments For Full Lifecycle Governance

- **ignite can be integrated with automated service provisioning.** Ignite's APIs for extracting artifacts from its repository can be invoked as part of automated continuous delivery systems, or be automated in more conventional ways via staging in configuration management, with scripted release into production. Ignite is designed to facilitate usage by firms that are pursuing DevOps (closer integration of Development and Operations). ICS Connect also offers tight integration with service registry/repositories, for firms that continue to pursue this approach.

- **ignite can integrate with service management tools for runtime metrics.** Customers have used ICS Connect APIs to integrate with service management tools such as those from HP or BMC, capturing "actuals" from the runtime to compare to SLAs and design goals defined when designing services in ignite. As this information accumulates and as analysts review new requirements, they can consider the impact new requirements may have on production system performance or capacity, and take proactive steps to shape demand and capacity to stay within required SLAs.

Bottom Line: As firms continue with ignite implementation, the opportunity to obtain additional benefits from integrating ignite with the API lifecycle downstream will become increasingly compelling.

Conclusions

There are so many ways to take advantage of the capabilities of ignite that you should consider the trade-offs between how much to attempt at each implementation stage, vs making each stage consumable by the people who have to manage change and implement these integrated capabilities. However, after each stage of implementation, reflect on what you have achieved already, consider the large menu of possibilities that remain (as defined above), and steer a course for the next few stages of implementation that strikes the right balance between risk and reward. digitalML recommends that you:

1. **Focus first on API governance.** Teams often struggle to move from the status quo to a more governed approach, which may be quite different from the way they worked before. Getting this right lays the foundation for doing more, otherwise you'll always be fighting a rear-guard action to establish the kind of portfolio your business requires.

2. **Exploit ServiceStore to bring managers into the fold.** Once managers start to see what they can learn from ServiceStore, they will want to encourage their teams to use ignite fully, so they can get more of this valuable information that they previously had no way to obtain.

3. **Start looking at how to automate or even replace specification documents.** Documents still play a central role in many firms' service development processes, because they enable multiple stakeholders to collaborate, albeit in a very 1980s fashion. digitalML believes there is great potential for additional benefit from replacing such practices with more streamlined online/real-time collaboration, based on what we've seen leading customers achieve already.

4. **Investigate the potential of other integration options.** After getting these more urgent requirements out of the way, it will be a good time to consider further options such as Project Management, DevOps, testing, and runtime metrics.