

WHITEPAPER_

Tech Strategy.

OUR TECH STACK

This document describes the technologies and methodologies used by Intelygenz in both internal and consulting projects. It is updated annually or whenever there is a significant change in the company's work areas.

SUMMARY

This document is divided by the different technical areas Intelygenz works in.

AREAS	CONTENT	STATUS
Architectures	Event Sourcing, Microservices	Core
Software Architectures	Layered, Clean/Hexagonal	Core
Backend	Java, NodeJs, Python	Core
Frontend	React, Angular, Vue	Core
Mobile	Android (Java, Kotlin), IOS (swift, Objective-c), React Native (Javascript)	Core
Machine Learning	PyTorch, SkLearn	Core
Platforms	Kubernetes	Core
Clouds	AWS, GCP	Core
Project Management	Scrum, Kanban	Core
Technical Methodologies	DevOps, TDD, BDD, ATDD, Infrastructure as code	Core

01.1

TECHNICAL AREAS: SYSTEM ARCHITECTURES

At Intelygenz, we use a range of system architectures, carefully selecting which to use for each project based on the objective we're trying to achieve. Some of our architecture patterns include:

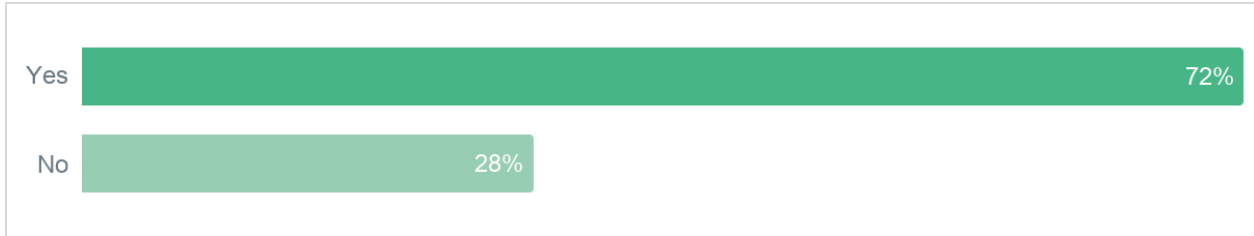
ARCHITECTURE	DESCRIPTION	STATUS
Event Sourcing	<p>In this pattern, you don't store the current state of your model in the database. Instead you store a series of state changes a.k.a events in an event store.</p> <p>Better fits in highly distributed systems where scalability and communications between microservices are a concern.</p>	Core
Microservices	<p>When you write your application as a set of microservices, you're actually writing multiple applications that will work together.</p> <p>This is our default approach when planning a new architecture as splitting the responsibilities inside the service is always a good take away.</p>	Core
Reactive	<p>Streaming async non-blocking architectures that provide great resource usage, fast response times and great scalability.</p> <p>In scenarios where the performance and processes parallelization makes the difference, reactive stacks are our preferred choice.</p>	Adopting
CQRS	<p>The central concept of this pattern is that an application has read operations and write operations that must be separated.</p>	Adopting

	When information availability is a concern, CQRS architecture fits perfectly, as you can have dedicated read models for specific use cases.	
SOA	In a Service Oriented Architecture (SOA) approach, multiple services communicate over a shared Enterprise Service Bus (ESB).	Leaving

01.1

TECHNICAL AREAS:
SYSTEM ARCHITECTURES

PROJECTS IN WHICH INTELYGENZ DESIGN & DEVELOPS SOFTWARE ARCHITECTURE



01.2

TECHNICAL AREAS: SOFTWARE ARCHITECTURES

When it comes to software, we're focused on building evolutionary architectures using the following pillars:

SOFTWARE ARCHITECTURE	DESCRIPTION	STATUS
Layered	The idea is to split up your code into "layers", where each layer has a certain responsibility and provides a service to a higher layer.	Core
Clean/Hexagonal	This approach isolates the central logic (core) of your application from outside concerns.	Core
Vertical	The code in a vertical architecture is separated by features, with either minimal or no code being shared between multiple application layers.	Adopting

01.3

TECHNICAL AREAS: DATABASES

Choosing the right place to store your data is crucial for the success of the project.

TECHNOLOGY	DESCRIPTION	STATUS
SQL	MySQL, DB2	Core
NoSQL	Couchbase, MongoDB	Core
Graph	Dgraph	Adopting
Search Engine	Elasticsearch	Core
EventStore	Kafka	Adopting

01.4

TECHNICAL AREAS: BACKEND

Intelygenz has a long portfolio of backend projects, which have been developed using the following programming languages:

LANGUAGE	FRAMEWORK	STATUS
Java	Java > 8. Spring Boot	Core
	Java > 8. Micronaut	Adopting
	Java 8, JSF, OSGI, Spring 5	Leaving
Python	Python 3, Flask, django	Core
	Python 2	Leaving
Javascript	NodeJS 10, Typescript 3	Core
GoLang	1.12>, No frameworks	Adopting

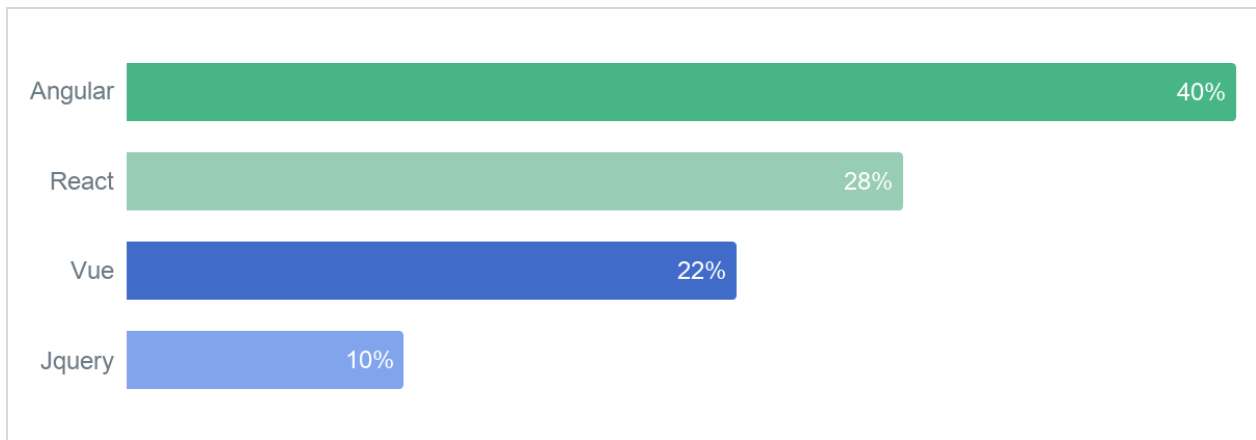
01.5

TECHNICAL AREAS: FRONTEND

When it comes to frontend, Intelygenz has developed projects with the following:

LANGUAGE	FRAMEWORK	STATUS
Javascript	React, Angular > 7, Vue	Core
	Jquery, Angular < 7	Leaving

THESE ARE THE FRAMEWORKS USED IN PROJECTS WHERE INTELYGENZ DEVELOPS FRONTENDS



01.6

TECHNICAL AREAS: MOBILE

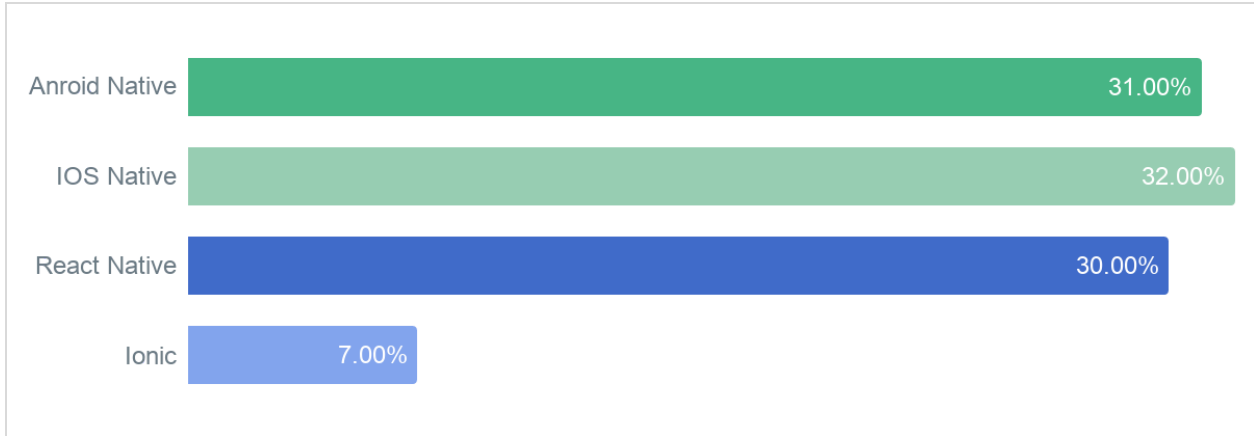
As mobile technologies play an increasingly larger role in our lives, Intelygenz continues to develop many applications using these four technologies:

PLATFORM	DESCRIPTION	STATUS
Android	Java, Kotlin	Core
IOS	Swift, Objective-C	Core
React Native	Javascript	Core
Ionic	Javascript	Leaving

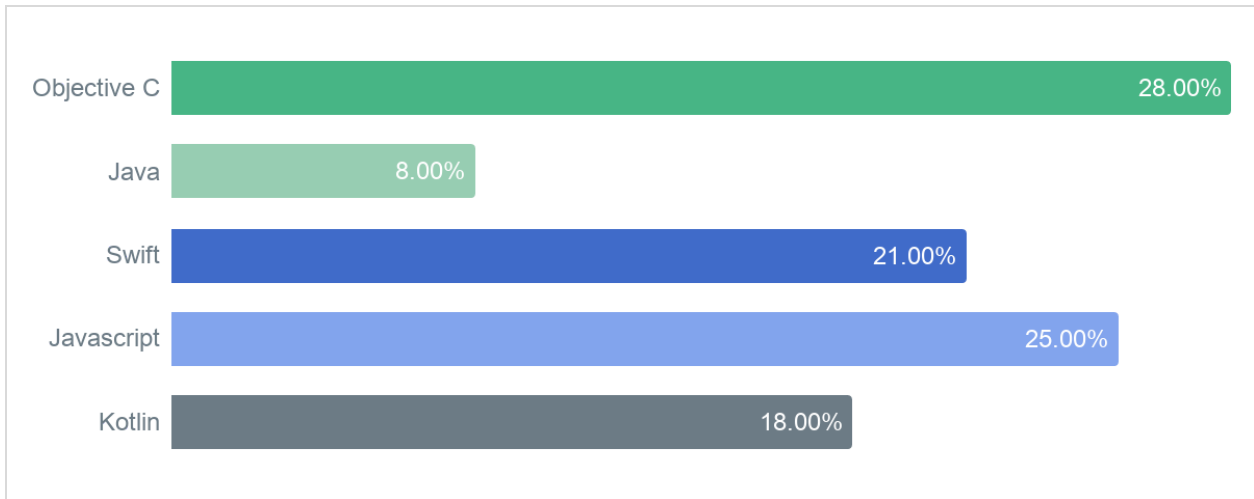
01.7

TECHNICAL AREAS: MOBILE

MOBILE PLATFORMS



MOBILE LANGUAGES



01.8

TECHNICAL AREAS: MACHINE LEARNING

Machine Learning (*) and Deep Learning (*) are currently the fastest changing areas in software. We opened our ML area in 2012 at our San Francisco offices.

The biggest challenge in ML area is the ability to move from lab to production because the needs of integration, scalability and model accuracy maintenance on ML solutions are totally different from one project to another.

LANGUAGE	FRAMEWORK	STATUS
Python 3	PyTorch	Core
	SkLearn	Core
	Tensorflow 1.X	Leaving
GoLang	Gonum	Adopting

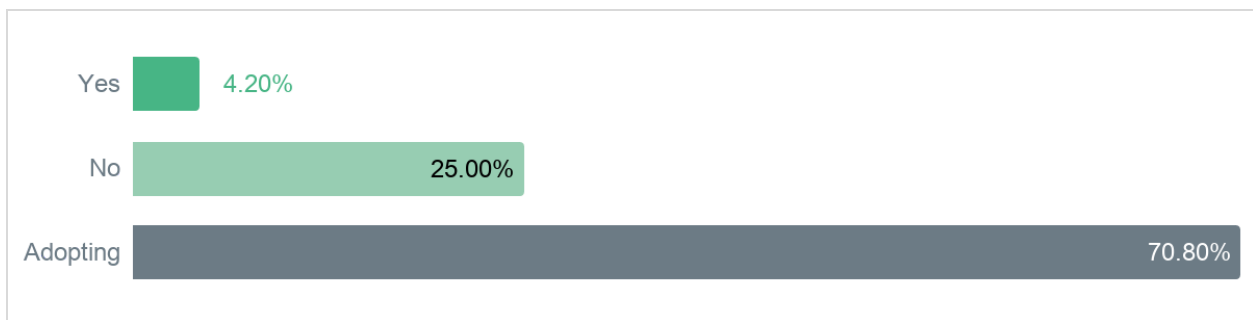
01.9

TECHNICAL AREAS: PLATFORMS

During recent years, we've worked on projects built on cloud native platforms, developing both cloud based and agnostic solutions:

PLATFORM	DESCRIPTION	STATUS
Kubernetes	Kubernetes, Amazon EKS / Google GKE	Core
Kubernetes	Azure AKS	Adopting
Google Cloud Run	Run stateless containers on a fully managed environment or in your own GKE cluster.	Adopting
AWS Fargate	Run containers without managing servers or clusters	Adopting
AWS ECS	High-performance container orchestration service that supports Docker containers and allows you to easily run and scale containerized applications on AWS	Adopting
GAE	Google App Engine	Leaving

THE CONTAINER ADOPTION ON INTELYGENZ PROJECTS ARE QUITE IMPRESSIVE:



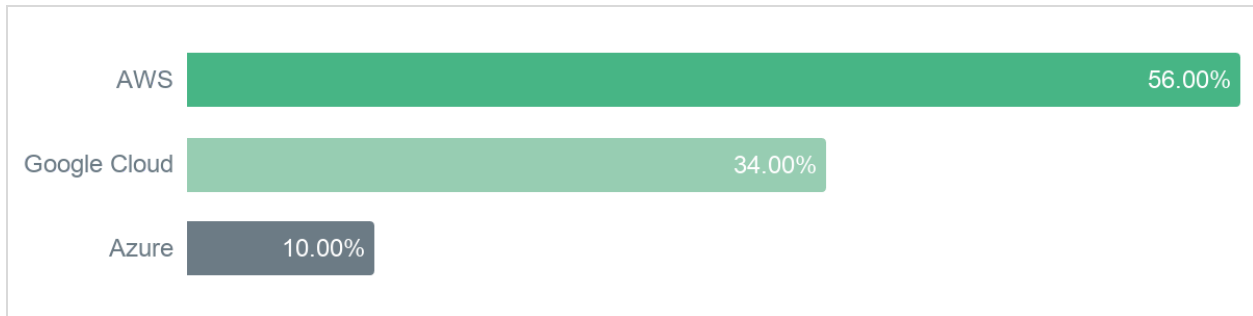
Projects at Intelygenz using containers

TECHNICAL AREAS: CLOUD

We've also developed platforms for the following public and private clouds platforms:

PLATFORM	DESCRIPTION	STATUS
AWS	Amazon Web Services	Core
GCP	Google Cloud Platform	Core
Azure	Microsoft Azure	Adopting

CLOUDS WHERE INTELYGENZ PROJECTS ARE DEVELOPING



02.0

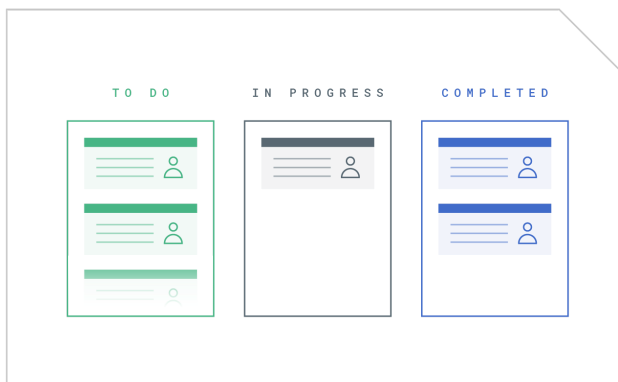
METHODOLOGIES

At Intelygenz, we strictly follow agile methodologies in both our project management and technical working practices.

02.1

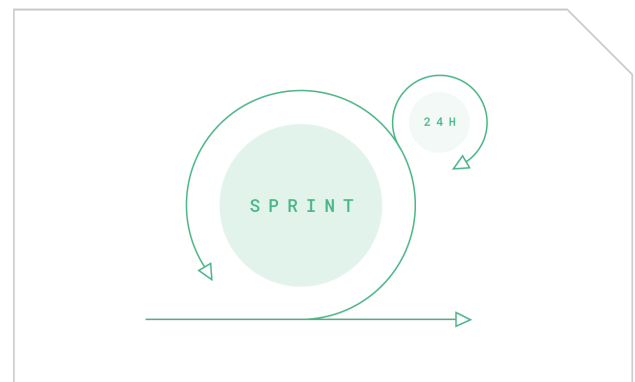
PROJECT MANAGEMENT

Within project management, we focus on two main methodologies:



KANBAN

Is a visual system for managing work as it moves through a process. Kanban visualizes both the process (the workflow) and the actual work passing through that process, restricting the maximum amount of work items in each stage. Most commonly adopted among platform/cloud teams.



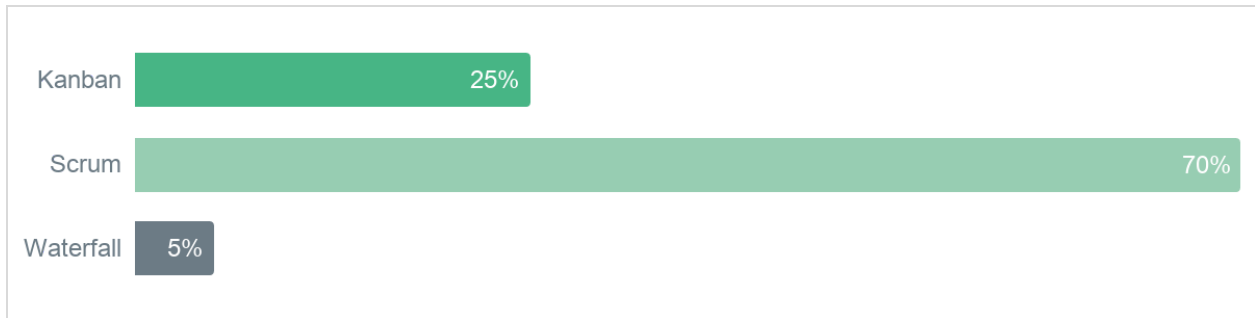
SCRUM

Is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, Scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve. Most commonly adopted among developer teams.

02.1

PROJECT MANAGEMENT

INTELYGENZ AGILE ADOPTION



02.2

TECHNICAL METHODOLOGIES

Our engineers use various technical methods that ensure their development is both safe and effective. These methods include:

METHODOLOGY	DESCRIPTION	STATUS
DevOps	DevOps (development and operations) is an enterprise software development phrase used to describe an agile relationship between development and IT operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between these two business units.	Core
TDD	Test Driven Development involves the repetition of a very short development cycle, where requirements are turned into test cases. The software is then improved to pass these new tests.	Core
BDD	Encourages teams to use conversation and concrete examples to formalize a shared understanding of how the application should behave.	Core
ATDD	ATDD is closely related to Test Driven Development (TDD), but stands out because of its highly collaborative approach. Acceptance Test Driven Development requires participation from customer-facing team members to help provide end user stories to the development/testing team. These stories are distilled into Acceptance Tests that guide the development process.	Core
Infrastructure as Code	Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.	Core
Immutable Infrastructure	Immutable infrastructure is all about immutable components which are recreated and replaced instead of updating after infrastructure creation. Immutable infrastructure reduces the number of places where things can go wrong.	Adopting

METHODOLOGY	DESCRIPTION	STATUS
Chaos Engineer	Chaos engineering is the discipline of experimenting on a software system in production in order to build confidence in the system's capability to withstand turbulent and unexpected conditions.	Adopting
GitOps	GitOps is a way to do Continuous Delivery. It works by using Git as a single source of truth for declarative infrastructure and applications.	Adopting
Shift Left Testing	Shift Left Testing moves testing and quality control nearer to the developer by automating and virtualizing environments in local machines. This way, the code leaves the developer's computer with less defects.	Adopting