BOHIOSYSTEM

SERVICE DATASHEET · BOHIOSYSTEM

Technology Architecture

We design the technical structure everything else is built on, and put in writing why each decision was made that way.

DOCUMENT
BS-SVC-02

VERSION
1.0

DATE
August 1, 2026

PREPARED BY
BohioSystem

THE SERVICE

> Architecture is the set of decisions that are **expensive to reverse.** This service exists to make them with judgment and document them, not to draw pretty diagrams.

We define how a system is structured: its components, how they communicate, where the data lives, how it scales, how it fails, and how it recovers. The result is not a diagram: it is a set of justified decisions the team can follow and question.

We work both on systems yet to be built and on existing platforms that need direction. In the latter case, the first deliverable is usually an honest diagnosis of what is there.

ADR	C4	No lock-in	Evolutionary
DOCUMENTED DECISIONS	DIAGRAM MODEL	VENDOR INDEPENDENCE	DESIGNED TO CHANGE

01 What we solve

The lack of architecture is not noticeable at first. It becomes noticeable once it is already expensive to fix.

COMMON SITUATION	WHAT IT MEANS FOR THE BUSINESS
No one knows why it was decided that way	Decisions live in the head of someone who is no longer there. Every change starts from zero, and mistakes that were already solved get repeated.
Everything is coupled to everything	A small change forces you to touch five modules and test the entire system. The cost of each improvement grows over time instead of shrinking.
It scales poorly and costs a lot	The only way to handle more load is to buy bigger servers, and the bill grows faster than the business.
A single point of failure	One component goes down and takes the whole operation with it, because no one ever designed what should keep working when something fails.
Locked into a vendor	The platform depends on proprietary services with no equivalent, so switching providers means rewriting.
Technical debates that never close	The team debates the same options every quarter because the conclusion and the reasoning were never documented.

02 What the service includes

01 Architecture design

Defining components, boundaries, contracts, and data flows for a new system or a major evolution.

Context and container view · Decomposition into modules · Contracts between components · Data model · Integration strategy

02 Existing architecture assessment

Diagnosis of a platform in operation: what is holding it up, what is slowing it down, and what is putting it at risk.

Map of the current system · Prioritized technical debt · Continuity risks · Bottlenecks · Phased remediation plan

03 Documented decisions (ADR)

Every relevant decision is recorded with its context, its alternatives, and its consequences. It is the project's technical memory.

Decision log · Alternatives evaluated · Consequences accepted · Status and review

04 Design for availability

We define what should keep working when something fails, and what each level of guarantee costs.

Recovery objectives · Redundancy by layer · Controlled degradation · Backup strategy · Recovery plan

05 Data architecture

Where each piece of data lives, who owns it, how it moves, and how it stays consistent across systems.

Conceptual and logical model · Sources of truth · Synchronization strategy · Retention and archiving

06 Technical governance

Rules that keep things consistent when several teams work on the same platform.

Development standards · Architecture review · Technology adoption criteria · Technical debt management

03 How we work

In short stages, with something verifiable at the end of each one.

PHASE	WHAT HAPPENS	WHAT YOU KEEP
01 · Discovery	We understand the business, the current state and the real constraint.	Diagnosis and agreed scope.
02 · Design	We define the solution, its risks and its execution plan.	Architecture and milestone plan.
03 · Build	We implement in verifiable increments, not in a single delivery.	Functional progress per milestone.
04 · Go-live	Controlled deployment, testing and knowledge transfer.	Solution in production and team trained.
05 · Evolution	Operation, measurement and continuous improvement under a service agreement.	Periodic report and living roadmap.

04 What you receive

Architecture document

— Context, container, and component views.
— Justified structural decisions.
— Quality attributes and their trade-offs.
— Explicit constraints and assumptions.

Decision log

— One ADR per relevant decision.
— Alternatives considered and discarded.
— Consequences accepted.
— Versioned format alongside the code.

Evolution plan

— Phases with dependencies between them.
— Risks per phase and their mitigation.
— Measurable success criteria.
— Effort estimate per phase.

Handover

— Review session with the technical team.
— Guide to reading the diagrams.
— Criteria for future decisions.
— Support during the first phase.

05 Technologies we work with

We choose the tool for the problem, not by trend or by agreement with a vendor.

C4 MODEL · ADR · ARC42 · UML · PLANTUML · STRUCTURIZR · EVENT STORMING · DDD · MICROSERVICIOS · MONOLITO MODULAR · ARQUITECTURA HEXAGONAL · CQRS · EVENT-DRIVEN · API-FIRST · WELL-ARCHITECTED

06 Engagement models

MODEL	HOW IT WORKS	WHEN IT FITS
Fixed project	Defined scope and deliverables, agreed price and timeline.	Bounded needs with a clear objective.
Monthly retainer	Reserved team capacity month to month.	Continuous evolution and sustained support.
Dedicated team	Profiles assigned exclusively under your direction.	Long programs or high workload.
Hour bank	Consumable hours with agreed priority.	Variable demand or one-off work.

07 Our commitments

COMMITMENT	HOW IT IS MET
Justified decisions, not imposed ones	Each choice is presented with its alternatives and consequences, so the client decides with full information.
Vendor independence	We do not recommend technology based on commercial arrangements. When we propose a proprietary service, we flag the cost of leaving it.
Proportional architecture	We do not design for a scale the business does not have. Overengineering is as costly as falling short.
Living documentation	Documents are delivered in version-controlled text format, alongside the code, so they do not go stale on a shared drive.

COMMITMENT	HOW IT IS MET
Real handover	The client's team comes out able to maintain and evolve the architecture without us.

WHEN THIS SERVICE MAKES SENSE ON ITS OWN

When decisions need to be made before building, when several teams need to coordinate on the same platform, or when an existing system has become expensive to change and an independent diagnosis is needed before investing in rewriting it.

08 Next step

GET STARTED

A one-hour session, at no cost, to review your current situation and define together the minimum scope to start with.

CONTACT
contacto@bohiosystem.dev

WEB
bohiosystem.dev

DOCUMENT
BS-SVC-02 · v1.0