# Bifrost

## The method

> *A method for engineering knowledge, small enough for a person, or an AI, to follow.*

Four layers, four links and one rule hold every requirement to its reason, its impact and its proof. The tools are built on it; the method comes first.



NEED · FUNCTION · SOLUTION · IMPLEMENTATION



INFO@ODINSINSIGHT.COM

# Four layers, and one rule.

Every requirement sits at one of four layers. All four describe the problem; none of them is the design.

01
## Need
What stakeholders, the business and the regulator require.
*Warn the occupants of a fire*

Fulfils ↑

02
## Function
What the system must be able to do.
*Sound the alarm*

Fulfils ↑

03
## Solution
What must be true. The stable anchor.
*At least 85 dBA at 3 m*

Fulfils ↑

04
## Implementation
How we build it. The layer that changes.
*Piezo sounder, part SND-12*

*Implementation is free to change. It must always still satisfy the layer above it.*

WHAT MUST BE TRUE · STAYS PUT

The receiver shall provide its current operating settings.

| GEN 1 | GEN 2 | OEM VARIANT |
|---|---|---|
| Settings as JSON over an RS-422 connector | A web interface on an embedded controller | Settings over Ethernet |

NEW SUPPLIER, NEW CHIP, NEW VARIANT: SWAP THE HOW. THE WHAT HOLDS.

THE LINKS

# Four kinds of link. Nothing else.

Small enough for a person, **or an AI**, to reason over.

### Fulfils
This satisfies that, across the layers: the reason-to-build chain.

### Decompose
This breaks into these parts, within one layer: system, subsystem, component.

### Bind
One choice here constrains a neighbour there. Collect them and the interfaces appear.

### Verifies
This test proves that requirement is met. Verified is computed, never set by hand.

## When something changes.

Change one requirement, and the links that leaned on it ask for a second look.

ONE CHANGE, GROUPED

"Run time revised: 16 links to re-check."

One notice per cause, not sixteen loose flags.

**Impact is a query.**

What depends on this, and what breaks if it changes, before you change it.

**Nothing is blocked.**

You decide when it is ready. Confirming is recorded: who, when and why.

**Proof on a changed line**

stops counting until someone confirms it still holds.

## Every requirement has a place.

Requirements hang on the architecture, so a question about a part is a question about its requirements.

### The system tree
Each element carries the requirements that apply to it.

### Interfaces appear
Bind links between neighbours are collected into the interfaces, not drawn by hand.

### One model
Architecture, requirements and proof. One ID per line, for life.

# A living library.

Reuse what your company already figured out, and keep the reasoning beside it.

### One library

Proven requirements your company owns. New projects start from what worked.

### Choose, and it follows

Pick wheels or tracks, and the rules that come with that choice arrive too.

### The record of why

Each decision, the alternatives it ruled out, and every requirement it created.

## Every release ends with a short review.

DRAFT → PROPOSED → **ACTIVE** → DEPRECATED

**Promote what you learned.**

A project's lesson enters as Proposed, and one named person accepts it.

**Retire what is no longer true.**

Deprecated, never deleted: its history and its traces stay.

## Constraints come from everyone.

FINANCE

Unit cost stays under €4,200 at 500 units.

SUPPLY CHAIN

Use the aluminium we already stock.

MANUFACTURING

Every fastener reachable with a standard hex key.

Each one owned by a named source and visible from day one, not found late in a review when change costs most.

# Approved work changes under review.

What has been approved is never edited in place. A change is proposed, reviewed and merged, and the record keeps all three.

**Baselines.**
Immutable snapshots, compared line by line.

**Change requests.**
Many edits in one request, shown word by word, with the impact first.

**Back to draft.**
Any change to an approved requirement sends it back for approval.

# Either seat of a contract.

Lines go down. Positions and proof come back.

THE CUSTOMER
Issues the specification, with the test method required on each line.

THE SUPPLIER
Answers every line: comply, partial or non-comply, and the case that will prove it.

IN BETWEEN
Their wording changes only through a change request, never a quiet edit.

# Proof that counts.

Every requirement names how it will be shown: inspection, analysis, demonstration or test. A case records the steps and the evidence, and the requirement is verified only when its cases pass.

*Answered means a position and a case. Never an approval somebody clicked.*

## A method AI can follow is the start of the AI context.

01 · METHOD
**A method AI can follow**
This brochure: four layers, four links, one rule.

02 · MODEL
**A model AI can read**
Every requirement with its reason, its place and its proof.

03 · STRUCTURE
**A record AI cannot quietly change**
Baselines and change requests. Agents propose; people merge.

04 · TOOLS
**Tools that put it to work**
Below.

## Where you use it.

AT THE DESK
### Bifrost Workbench
The whole method in one tool: model, verify, change, exchange, documents and agents.

FOR A SMALL TEAM
### Bifrost Lite
The method from the specification down, for a team answering a customer.

ON A PHONE
### Bifrost Pocket
Review, run a test and raise an issue at the bench.

### Norn
Derives the set from the Needs.

### Týr
Obligations and evidence.

### Fenrir
Failure analysis on the same graph.

### Mimir
Standards, clause by clause.

## Read the method. Then try it on your system.

The methodology is published. We teach it on your own requirements, then build the AI context with your team.

odinsinsight.com/methodology · info@odinsinsight.com

*The methodology documents and the Bifrost Product Description define the method and the binding scope.*