ODIN'S INSIGHT

The AI context for engineering.

AI can only do the engineering work it can see. We give it the method, the model, the structure and the tools, inside your programme.

The knowledge that outlasts your engineers is the same knowledge your AI needs. We make it explicit, once, for both.



METHOD · MODEL · STRUCTURE · TOOLS

ODINSINSIGHT.COM

INFO@ODINSINSIGHT.COM

You have said at least four of these.

The documents survive when an engineer leaves. The understanding does not.

"We found the missing requirement in integration."

→ Derivation proposes the set from the Needs, before build.

"Nobody can explain why this requirement exists."

→ Every constraint traces to the Need behind it.

"The threshold changed and nothing downstream moved."

→ Every dependent link is flagged until a person confirms it.

"We start every programme from a blank page."

→ A governed library carries proven requirements forward.

"The document took three weeks to assemble."

→ Documents are generated from the model.

"Which requirements changed since the last review?"

→ Immutable baselines, compared line by line.

AI can only do the work it can see. We build what it needs to see.

01 · Method

A method AI can follow

Four layers, four links, one rule, and quality rules any reviewer checks against.

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, change requests and approval. Agents propose; people merge.

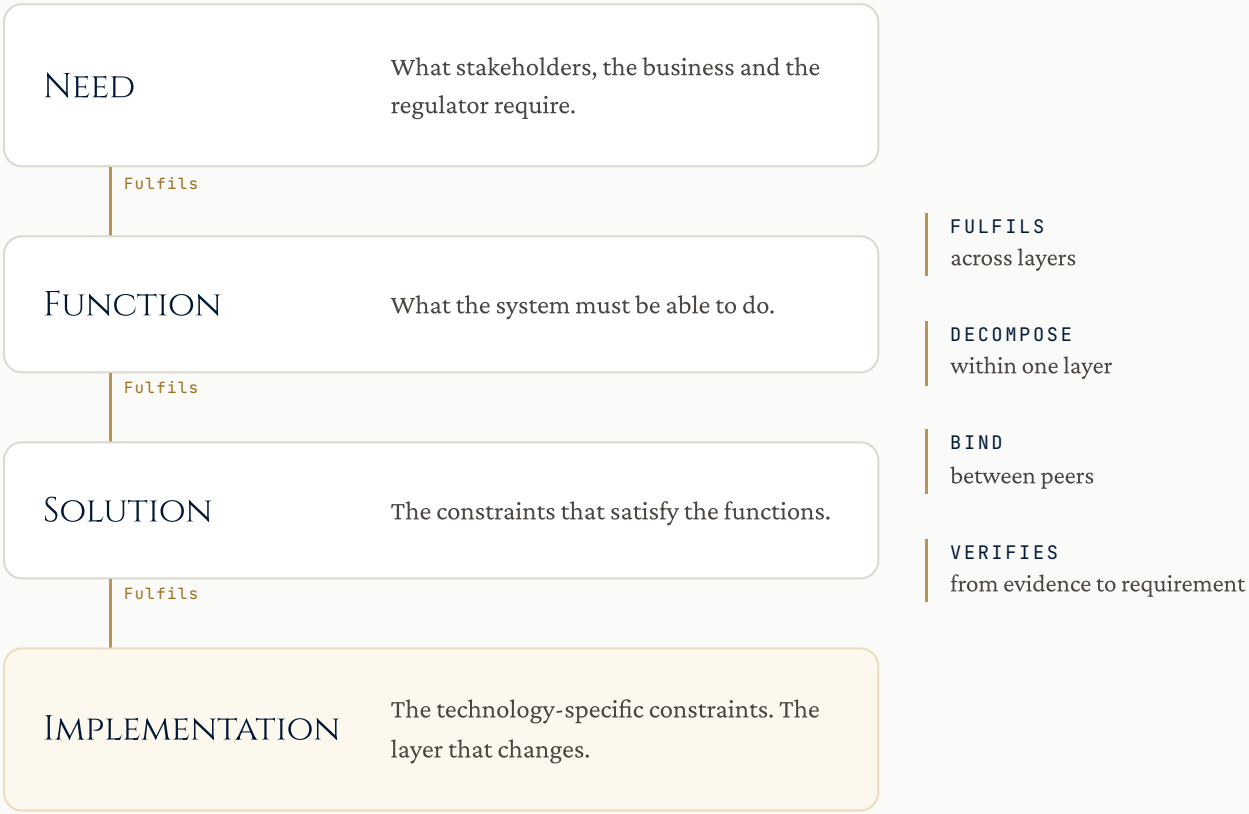
04 · Tools

Tools that put it to work

The Workbench and Pocket, agents over MCP and Huginn, customer mails and sheets in, SysML v2 and documents out.

Four layers, and one rule.

Every constraint in your system sits at one of four layers. All four describe the problem; none of them is design.



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

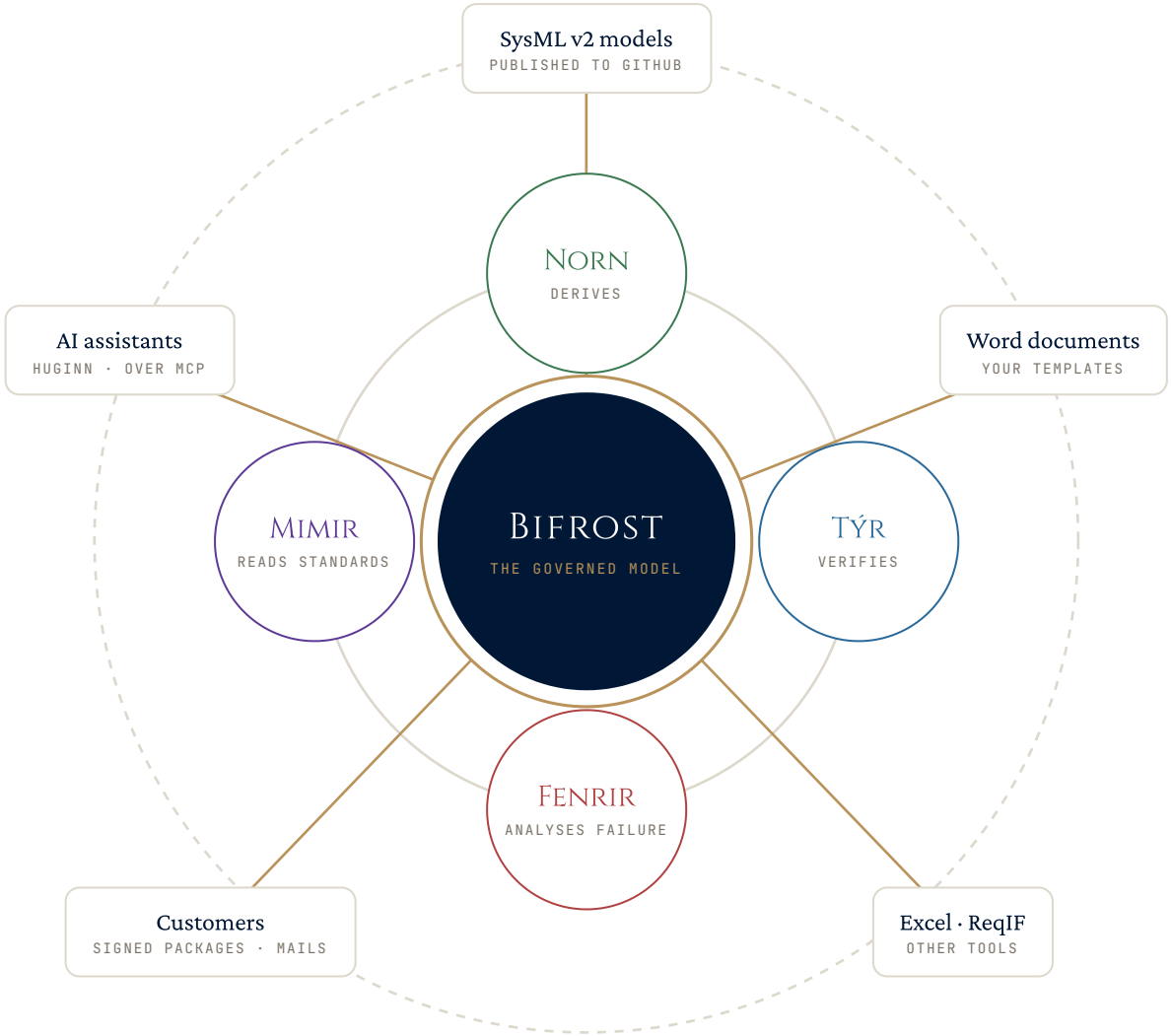
A refresh, not a rewrite.
Change the part, prove it still satisfies the layer above.

Impact is a query.
What depends on this, and what breaks if it changes.

Fixed on purpose.
Every project uses the same model, so the trace can be analysed, not just shown.

One model in the middle. Everything around it.

Out in their format. Back in under review. One record of every exchange.



Bifrost

Four layers, the library, baselines and change requests.

Norn

Needs into Functions into Solutions, reusing what you own.

Týr

Obligations and evidence, so verified is computed.

Fenrir

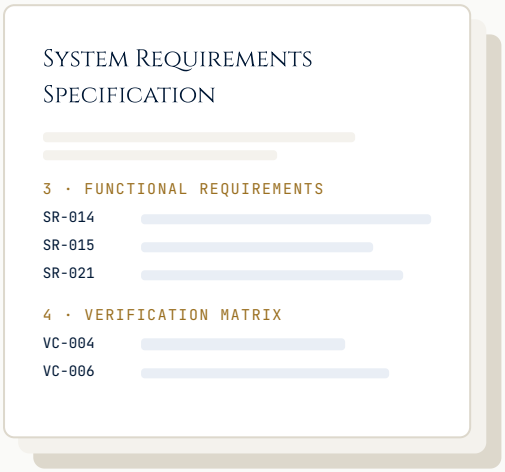
Failure analysis on the same graph; mitigations return as drafts.

Mimir

Standards turned into governed requirements, clause by clause.

The documents an auditor asks for.

In a regulated domain the document is what gets reviewed. Generated from the model, it is never out of date.



Documents as queries.

Each section is a live selection of the model, on your own Word template.

Traceability built in.

Trace and verification matrices and compliance responses.

Change you can show.

Baselines, change requests and the report between them.

Agents propose. People merge.

A language model is only as good as what it is given. The Bifrost model gives it the structure: why each requirement exists, what it fulfils and what proves it.

Your agent reads the model.

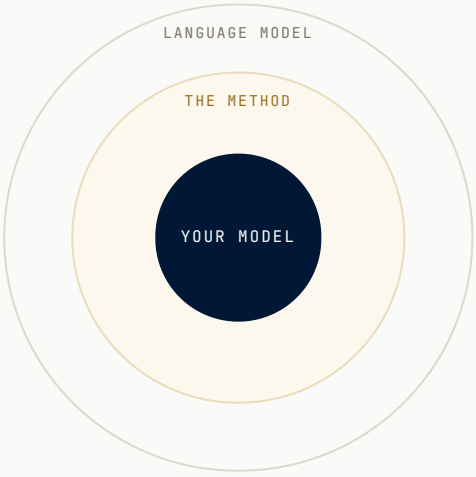
Bifrost serves the model and the method to AI agents over MCP. Huginn, our assistant on your own hardware, takes an issue like a colleague and drafts the change in your name, reads the customer's mails into proposed requirements, and builds the model from their sheet by asking the engineer.

It proposes, under review.

Each change an agent proposes arrives as a change request, shown word by word.

A person merges.

Your team merges or declines it. Norn derives the same way: proposals, never commits.



Yours, inside the fence.

One instance per customer.
Separated by deployment, not by a filter in a shared database.

Air-gap-true.
Retrieval stays local. Generation can be switched off or run inside your network.

Turnkey.
Application, database and services ship as one unit. Nothing phones home.

How we work with you.

01
Assess
A short diagnosis that scopes what follows, and says so when we are not the answer.

02
Teach
Training on your own systems: MBSE, SysML and the method.

03
Deploy
We build the AI context inside your programme: method, model, structure and tools. We stay until your team runs it.

We build the AI context for your engineering, then hand it to your team.

Start with an honest assessment.

Tell us the programme and the seam you are worried about, and we will tell you whether we are the right group for it.

info@odinsinsight.com · odinsinsight.com

The Product Descriptions define the binding scope.