Bifrost Workbench

> *Where the method gets used: model, verify, change and deliver, in one place.*

The Bifrost method needs a tool your engineers will open every day. The Workbench is that tool, in the browser, on your own instance, with your agents working beside your team.



MODEL · VERIFY · CHANGE · CONNECT

AT A GLANCE

Bifrost Workbench in one page.

One model of your system, from Need to proof, that the whole team works in, and that every document, review and exchange is drawn from.

Model and write — P. 03
Four layers and four links on every row; the chat beside every screen.

System context — P. 04
What sits outside the system, and the Functions written between them.

Verification — P. 05
Cases, recordings and evidence; the trace matrix shows what is proven.

Document generation — P. 06
Documents defined as queries and rendered to Word on your templates.

Change management — P. 07
Approved lines change only through a reviewed, approved request.

Intake and the Interview — P. 08
The customer's mails read for you; their sheet built into a model by asking you.

Issue management — P. 10
Questions pinned to the requirement or part, planned against milestones.

SysML v2 on GitHub — P. 11
Baselines publish as pull requests; edits come back under review.

Agents and Huginn — P. 12
Your AI agent over MCP, and a shared assistant on your hardware. They propose; a person merges.

Lite and Pocket — P. 12
The same model for a small team, and on a phone at the bench.

Less rework.
Gaps, suspect links and missing proof show before a review finds them.

Audit-ready.
Every change has a request, a reviewer and a reason. Nothing approves itself.

Ready for AI.
A structured, governed model is the context an agent needs to help safely.

MODEL AND WRITE

One model, every view.

Tables, trees, the Board and the matrix are all the same model. Change it anywhere and every view agrees.

01 **MODEL**
Four layers and four links, traced on every row.

02 **ARCHITECTURE**
The system tree, with each element's requirements on it.

03 **WRITE**
The assistant checks and drafts; you decide.

04 **SHAPE**
The DSM proposes how a node splits.

05 **REUSE**
Start from the library; Norn derives a draft set from your Needs.

06 **ASK**
The chat beside every screen knows what the screen shows.



THE BOARD · DRAWN FROM THE WORKBENCH · ILLUSTRATIVE PROJECT

Saved views.
Filter once, name it, share it with the team.

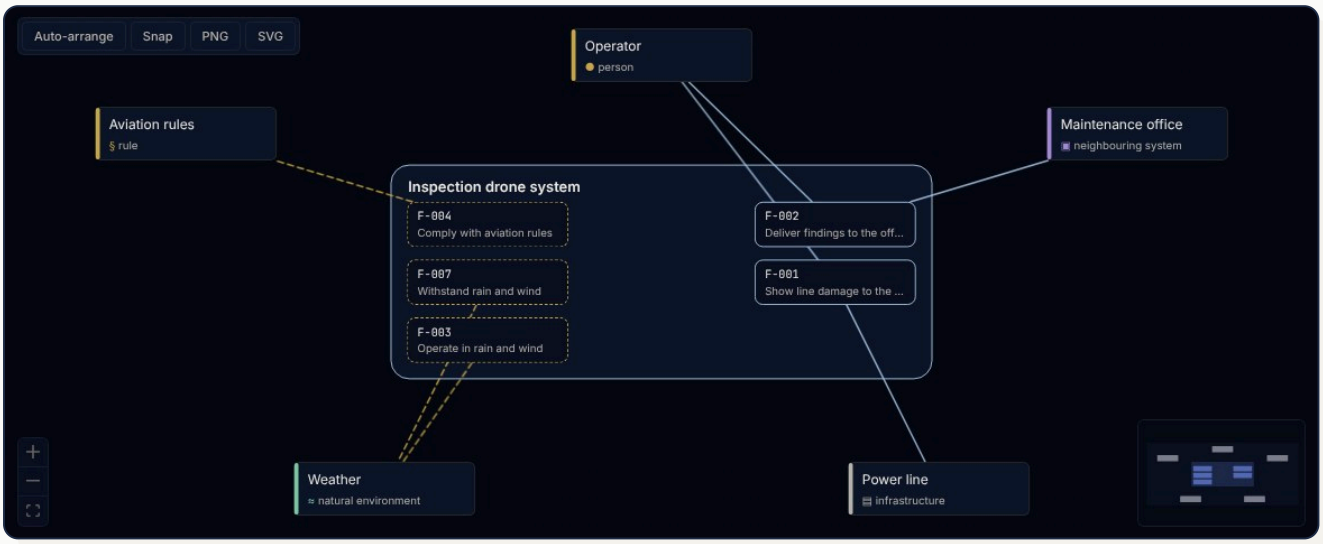
Every edit recorded.
Each line keeps its history: who, when and what changed.

Keyboard first.
Search or command from anywhere, by ID or by word.

SYSTEM CONTEXT

Start from what sits outside.

Before Needs become Functions, list what the system meets: people, neighbouring systems, the weather, the infrastructure and the rules. Each Function names the elements it acts between.



THE CONTEXT DIAGRAM: THE SYSTEM HOLDING ITS FUNCTIONS, ITS ENVIRONMENT AROUND IT · REAL SCREEN, DEMO PROJECT

Elements and Needs.

People, flows, neighbouring systems, the natural environment, infrastructure and rules; each Need names the ones it concerns.

The Function window.

Functions written Need by Need: Norn proposes one for each pair of elements; correct it, accept it, or link one you already have.

Coverage shows the gaps.

A Need nothing serves, an element no Need names, a pair with no Function between them.

A diagram you can arrange.

Drag, align and save it for the whole team; PNG and SVG for a review pack.

One level down.

A subsystem's context: its sibling nodes, and what its parent acts between.

Norn proposes.

Elements and Functions are proposed; nothing is added until you accept it.

Into SysML v2.

The context exports with the model: elements as parts, Functions naming them.

VERIFICATION

Verification, traced to every line.

Every requirement names how it will be proven. Cases carry the steps, runs carry the evidence, and the trace matrix shows the whole chain at a glance.

VERIFICATION · REPORTS (RVTM)

REQUIREMENT	STATEMENT	CASE	METHOD	LATEST RUN	VERIFIED
SR-014	The alarm shall sound at least 85 dBA at 3 m.	VC-003	Test	Pass	✓
	1. Trigger the alarm in the chamber · Expected: ≥ 85 dBA at 3 m			88 dBA	
SR-021	The detector shall report a fault within 24 h.	VC-007	Demonstration	Fail	×
SR-030	The housing shall be flame retardant.	VC-011	Inspection	Not run	–
SR-033	Battery life shall be at least 10 years.	–	Analysis		No case

THE TRACE MATRIX · DRAWN FROM THE WORKBENCH · ILLUSTRATIVE PROJECT

Coverage matrix.

Every case with what it verifies and its acceptance criteria; gaps show before a review does.

Cases and steps.

Inspection, analysis, demonstration or test, with procedure documents attached and steps mapped to requirements.

Recordings you can stand behind.

One per variant, on a named unit with its serial number, from the desk or a phone at the bench. Locked once recorded; corrected with a reason.

Verified is computed.

Only approved cases that pass count. A result an assistant records waits for a person to confirm it; change the line, and its proof waits too.

RVTM to PDF.

The requirements verification traceability matrix, with each case's steps and latest results.

Compliance matrix.

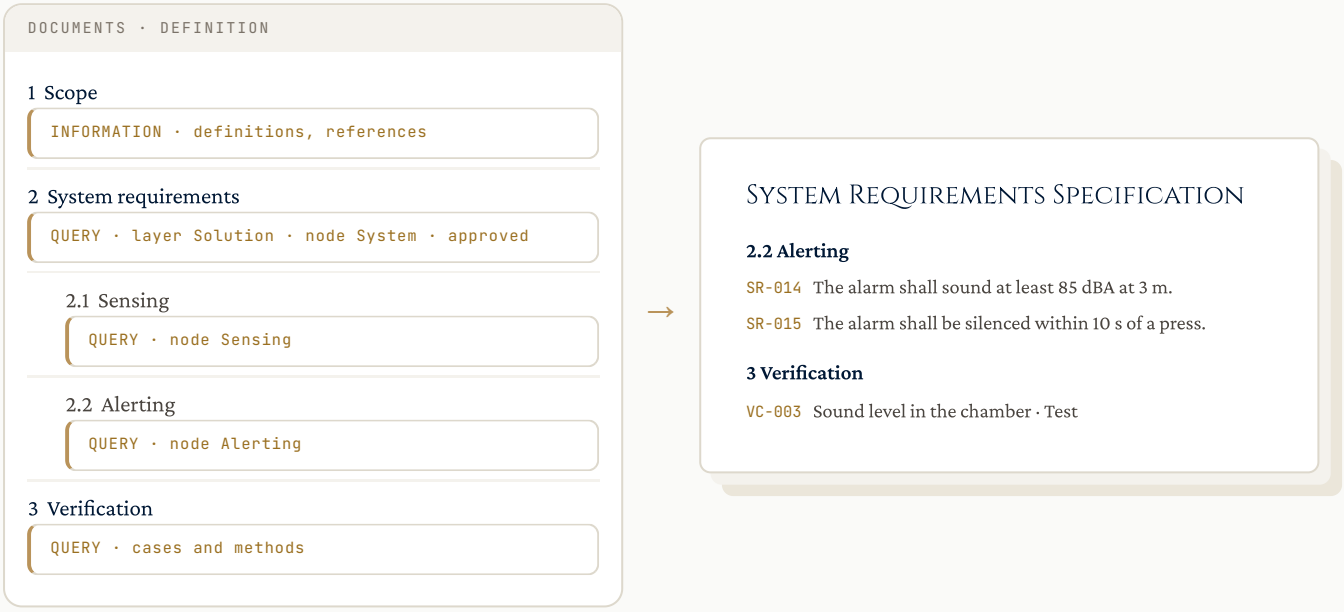
Comply, partial or non-comply on every customer line, with the case that proves it.

Trace and impact.

Follow any line up to its Need and down to its parts and proof.

Documents written by the model.

In a regulated domain the document is what gets reviewed. Define it once as a set of queries; the Workbench renders it to Word on your own template, from the current model or any baseline.



A DOCUMENT DEFINITION AND THE PAGE IT RENDERS · ILLUSTRATIVE

Sections are queries.

Each section is a live selection by layer, status, node or attribute, or pinned lines, with a preview of what it will render.

Seed from architecture.

The outline builds itself from the Solution tree; add definitions, abbreviations and assumptions as information elements.

Your Word templates.

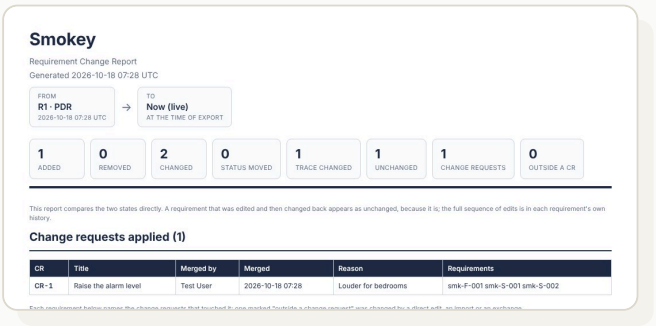
Managed by an admin, with a built-in default. Never out of date, because it is never typed.

Every other format.

Excel, CSV, an HTML report and ReqIF from the same selection; a portfolio status pack in one file.

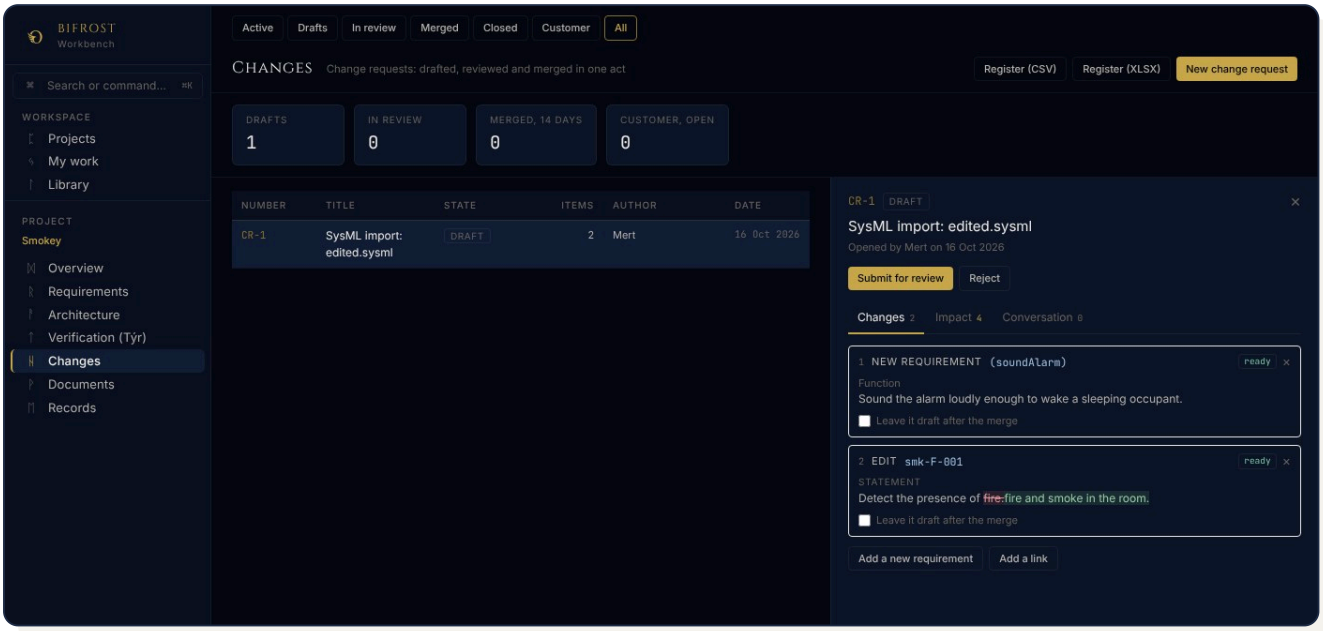
Reports for an auditor.

The change report between two baselines, and the register of every change request: who raised, approved and merged it, and why.



Change, under review.

Approved work is never edited in place. In Changes it is proposed, reviewed and merged, like a pull request for your requirements, and the record keeps all three.



A CHANGE REQUEST, ITS EDITS SHOWN WORD BY WORD · REAL SCREEN, DEMO PROJECT

Many edits, one request.

Text, attributes, links and cases together, shown word by word.

Impact first.

What depends on each edit, before anyone approves it.

Review and merge.

Comment, ask for changes or approve. Only an approved request merges, and every approval is a click of its own, whatever the role.

Protected lines.

Switch protection on and approved lines change only through a request.

The customer answers.

A change to their line goes out in order: approved, asked, sent, then accepted or rejected. Every round is kept.

The letter's attachment.

One export gives what is approved and not yet sent, ready to print to PDF. Nothing internal is in it.

Baselines.

Immutable, compared line by line, grouped by the request that moved each line.

Audit trail.

Every merge in each line's history links its request.

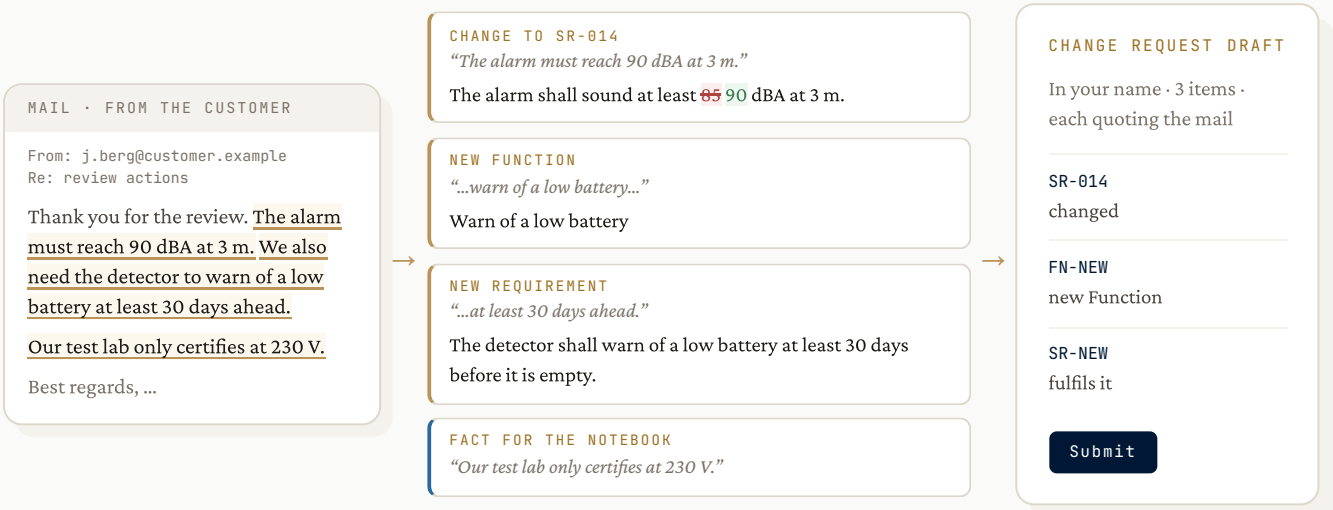
From anywhere.

Imports, exchanges, Intake and agents all arrive as requests. A merged one can be reverted.

INTAKE

From the customer's mail to a change request.

Customers write in mails, meeting notes and documents. Drop them on Intake: the assistant reads them and proposes the requirements inside, each beside the customer's own sentence. You decide on every one.



MAIL IN, PROPOSALS TO DECIDE, ONE DRAFT REQUEST OUT · ILLUSTRATIVE



TWO SENTENCES OF ONE MAIL, WRITTEN INTO ONE CHANGE TO ONE LINE · REAL SCREEN

Drop it or paste it.
A saved mail, a Word document, a PDF or plain text. On site, a mailbox rule can feed the assistant's folder, or forward the mail to the assistant's own Intake address.

Nothing invented.
A proposal whose sentence is not in the text, or whose value the sentence does not carry, is dropped and counted.

Changes, not duplicates.
A line you have shows now beside proposed. Two asks for one line become one change, or a conflict you settle.

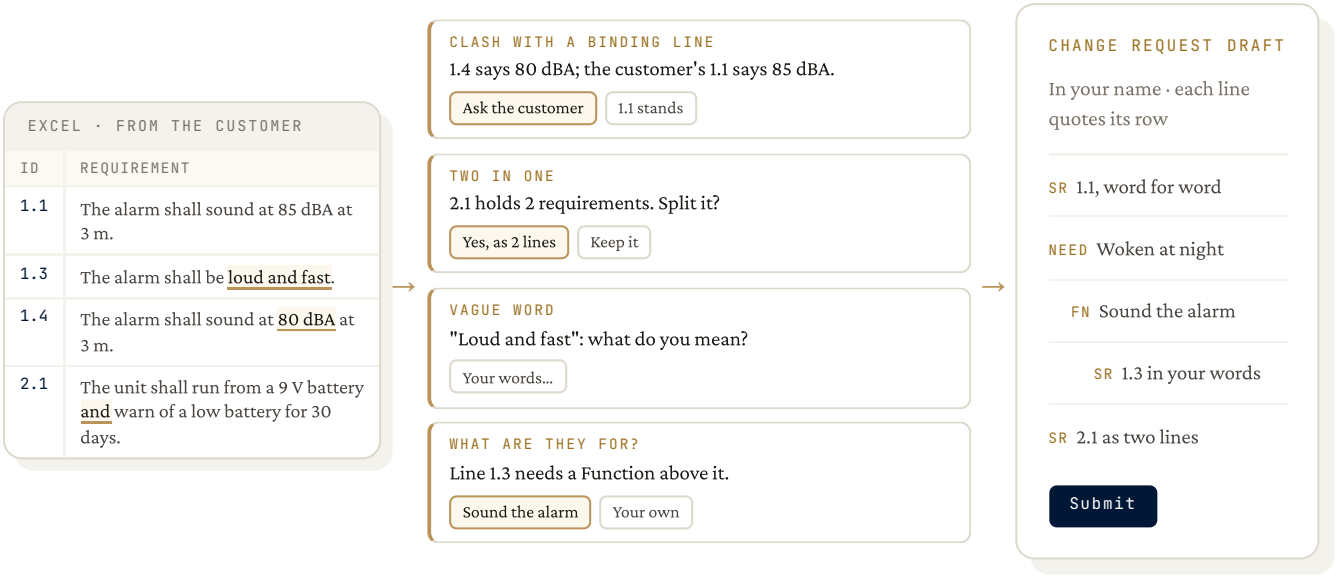
The sender counts.
A mail domain names the stakeholder; a line from the customer starts contractual.

You decide.
Accept, edit or reject each one. The accepted ones become one draft request in your name.

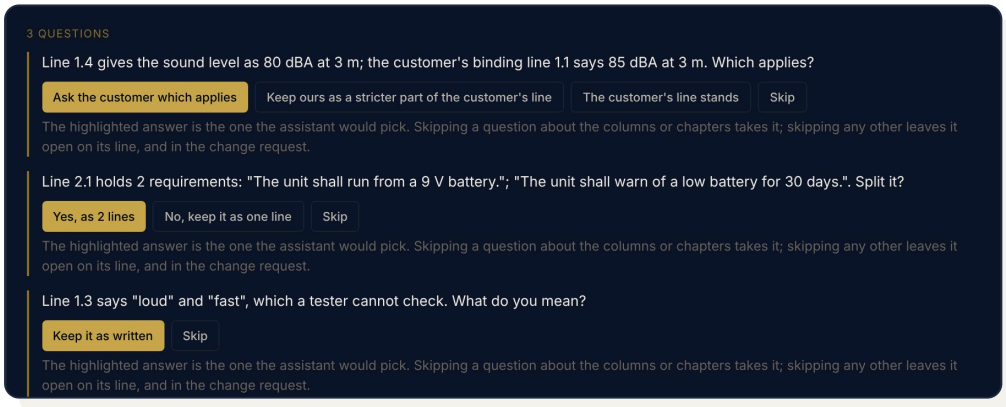
On your hardware.
Huginn reads it on site, with a local model if nothing may leave your network.

From the customer's sheet to a model.

A specification often arrives as a spreadsheet or a Word table. Drop it on Intake: the assistant reads it, asks you a stage at a time, and writes what you settled into one draft change request.



A SHEET IN, QUESTIONS TO ANSWER, ONE DRAFT REQUEST OUT · ILLUSTRATIVE



THE ASSISTANT'S QUESTIONS; THE HIGHLIGHTED ANSWER IS THE ONE IT WOULD PICK · REAL SCREEN

A stage at a time.

The columns and chapters, then the whole sheet, then each chapter's lines. Answer, type your own words, or skip.

The customer's words stay.

Lines you mark binding enter word for word. A clash with one becomes a question for the customer.

Checked, not trusted.

A split may not add or lose a number; a Need is proposed only from the sheet's own sentence.

Good enough.

A bar per chapter shows when the model is complete enough to write.

Live.

While the page is open, the next question comes within seconds.

Huginn onboarding.

It asks four questions about a project and drafts its description for you to accept.

ISSUE MANAGEMENT

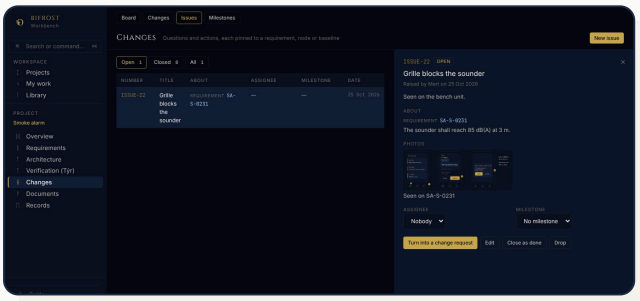
Issues, pinned to the model.

A question or an action belongs next to the thing it is about. Issues pin to a requirement, a part or a baseline, and the Board shows everything waiting in the project.

CHANGES · BOARD · GROUPED BY MILESTONE: CDR

TO DO	IN REVIEW	CHANGES ASKED	READY TO MERGE	DONE
ISSUE-22 Grille blocks the sounder	CR-9 Raise the alarm level	CR-8 Self-test interval	CR-7 Housing material	ISSUE-19 Chamber booked
ISSUE-24 · AI agent Battery spec cites an old standard				

THE BOARD: CARDS MOVE WHEN THEIR STATE DOES, NOTHING IS DRAGGED · ILLUSTRATIVE



AN ISSUE RAISED AT THE BENCH, WITH PHOTOS · REAL SCREEN

Raise it where you are.

From a requirement, the Issues view, or a phone at the bench with a photo.

Assign and plan.

An assignee and a milestone; comments keep the answer with the question.

Close it, or change it.

Done with what answered it, dropped with a note, or turned into a change request in one step.

Milestones.

PDR, CDR, a contract issue: open requests and issues, and the days left.

Cut the baseline.

A milestone captures its baseline and records what was still open.

Agents too.

An AI agent can raise and comment on issues, marked as its own.

Your model, as SysML v2 on GitHub.

Link a project to a folder of a GitHub repository. Each baseline publishes as a SysML v2 pull request; modellers edit in their own SysML editor, and their changes come back under review.

01 **Baseline** — *captured in Bifrost*

→

02 **Pull request** — *SysML v2, opened on GitHub*

→

03 **Merged and tagged** — *your CI parses it first*

↓

04 **SysML editor** — *modellers edit*

←

05 **Import preview** — *a check on the pull request*

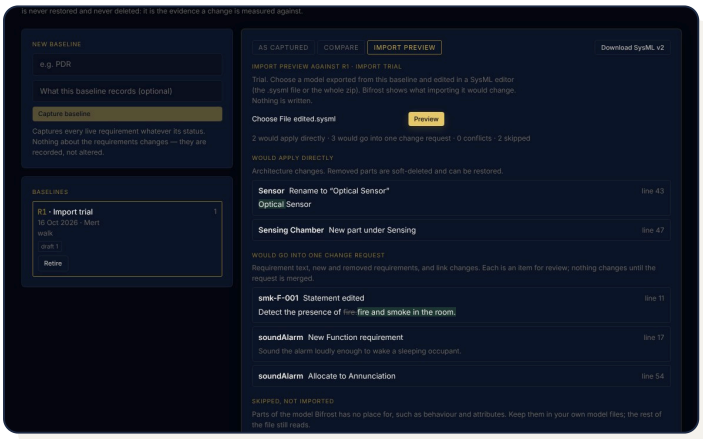
←

06 **Change request** — *reviewed, merged*

↑

BIFROST STAYS MASTER

architecture applies · requirements are reviewed



IMPORT PREVIEW: WHAT APPLIES, WHAT GOES TO REVIEW · REAL SCREEN

Publish as a pull request.

One branch per baseline; a person merges it on GitHub, and Bifrost tags the merge.

See the checks.

The baseline shows its pull request, open or merged, and whether the model parses.

Import from GitHub.

Pick a pull request or branch. Architecture applies; everything else becomes one change request.

Your own App.

Each instance connects with its own GitHub App, never a shared key.

Other tools.

Excel, CSV and ReqIF in and out, with a preview before anything is written.

Partner instances.

Signed exchange packages carry lines between customer and supplier.

Agents propose. People merge.

Connect your own assistant.

Your AI agent reads the model and the method over MCP. Its edits arrive as one change request beside your own; it raises issues and comments on them too.

Huginn, a colleague for the team.

One shared assistant on your own hardware. Assign it an issue and it drafts the change in your name; comment and it revises. It reads what arrives on Intake, interviews you about a customer's sheet, keeps the project's notebook, reviews overnight and writes each person a morning digest.

A person decides.

No assistant changes a status, submits or merges. A result it records waits for a person to confirm.



ILLUSTRATIVE

Three ways to work. One model.

AT THE DESK

Workbench

The whole method, from stakeholder Needs to parts, with Norn, the DSM, SysML v2 and GitHub.

FOR A SMALL TEAM

Lite

Answer a customer's specification: Solution and Implementation lines, verification and compliance.

ON A PHONE

Pocket

Review, run a test step by step and raise an issue with a photo, from the bench.

Lite becomes the full Workbench with one configuration change. No migration. Intake, the Interview, Pocket and agents work in both.

Your own instance.

Hosted, or on your own hardware, air-gapped if you need it.

Guide and support.

The manual ships with the version you run; report a problem from any screen.

English and Turkish.

Switch the interface in one click.

See it on your own specification.

We set up an instance, load your requirements and walk through the first ones with you.

Screens are from a demo project. The Bifrost Product Description defines the binding scope.