CASE STUDY · STORMS PETREL

DEEPSOFT



The optimizer already worked. It just did not live where the models live.

Production optimization with short term operational constraints, brought into Petrel without rewriting a single optimization routine.

CLIENT

Universidade de Brasília · Finatec

END USER

A Brazilian operating company

STACK

Ocean · C# · .NET · Python · Eclipse

THE SITUATION

DEEPSOFT

Every well control, on every date, inside the platform limits

Maximizing net present value means choosing rates, pressures, status and fluid mixing for producers and injectors over the whole production horizon, while honoring what the reservoir allows and what the platform can process, separate and inject on any given day.

In pre-salt carbonates the problem grows on every axis at once: many wells, long simulation horizons and injection strategies such as WAG.

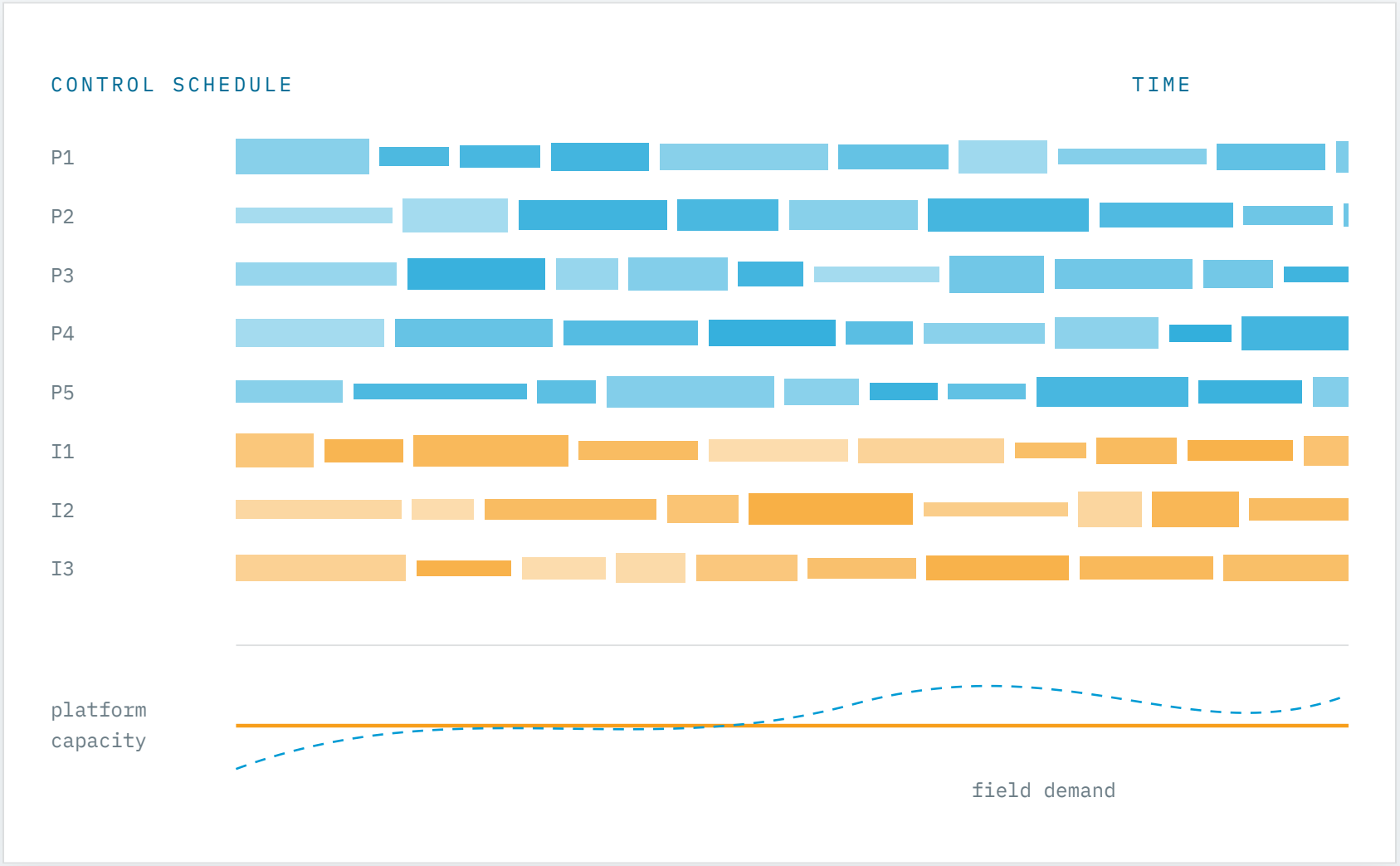


Fig. 01 · Schematic. Producer and injector controls per date, bounded by short term platform capacity.

DEEPSOFT

A working method, one platform away from being useful

The method existed

UnB had built an optimization methodology for pre-salt conditions and implemented it in STORMS, within a research project run with a Brazilian operating company.

It ran somewhere else

STORMS operated outside the geological and reservoir modeling environment, away from the models, cases and strategies the engineers actually work on.

Do not break maintainability

The original software had to stay maintainable and keep evolving in the hands of the research team that wrote it.

Do not duplicate the algorithms

Porting the optimization routines into Ocean would create a second copy of the science, and a second thing to keep in sync forever.

> The brief was integration, not reimplementation. That decision shaped everything else.

ARCHITECTURE

DEEPSOFT

Two programs, one workflow, no second copy of the science

The plug in embeds a Python interpreter and runs STORMS as a separate process. Data and commands travel over sockets. The optimization algorithms never leave STORMS.

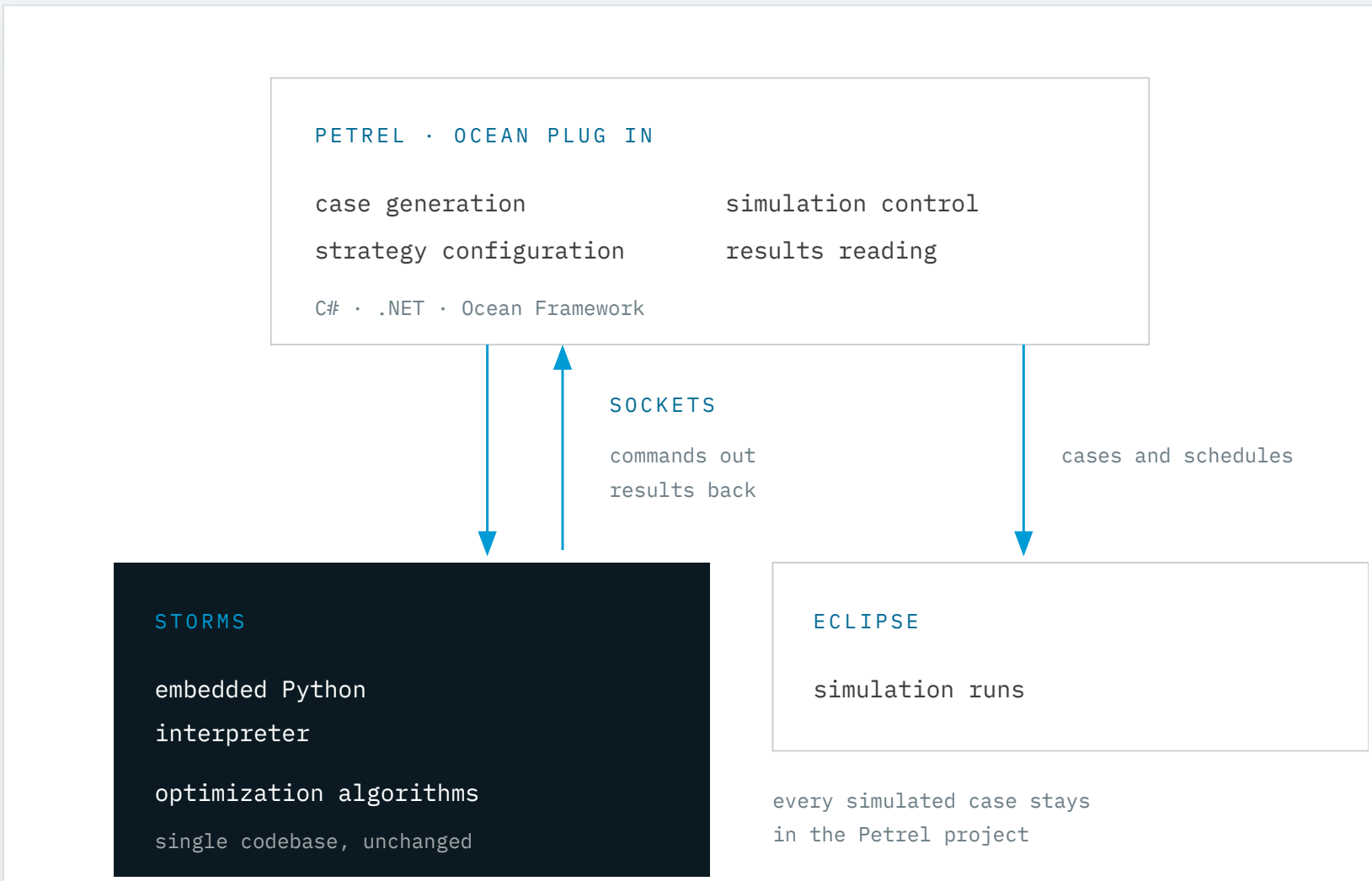


Fig. 02 · Schematic. Integration by process and sockets, chosen over porting the routines to C#.

The engineer configures a project, not a YAML file

Base case, starting point in time with restart support, controls and constraints per well and per date, schedule parameters, and the optimization algorithm, all from a dedicated interface inside Petrel.

DIRECT SIMULATION | NORMAL REACTIVE | ENHANCED REACTIVE | MNPV LIKE | SIMPLEX

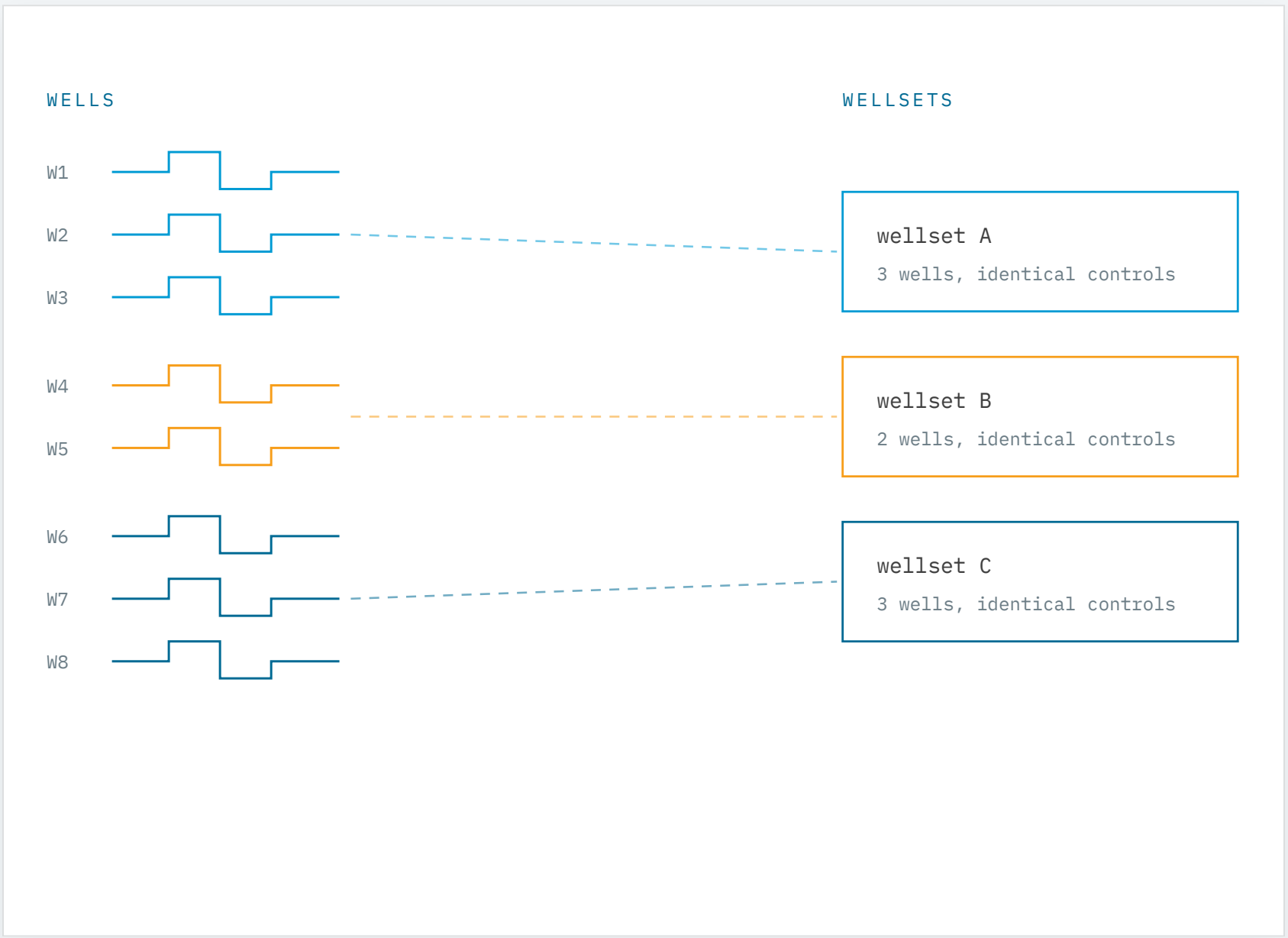


Fig. 03 · Schematic. Wells sharing identical controls are grouped automatically, which shortens strategy generation.

DEEPSOFT

You watch the optimization think

A dedicated progress window follows every iteration: cumulative NPV, total field injection and production rates, the performance of each well and a complete log of the process.

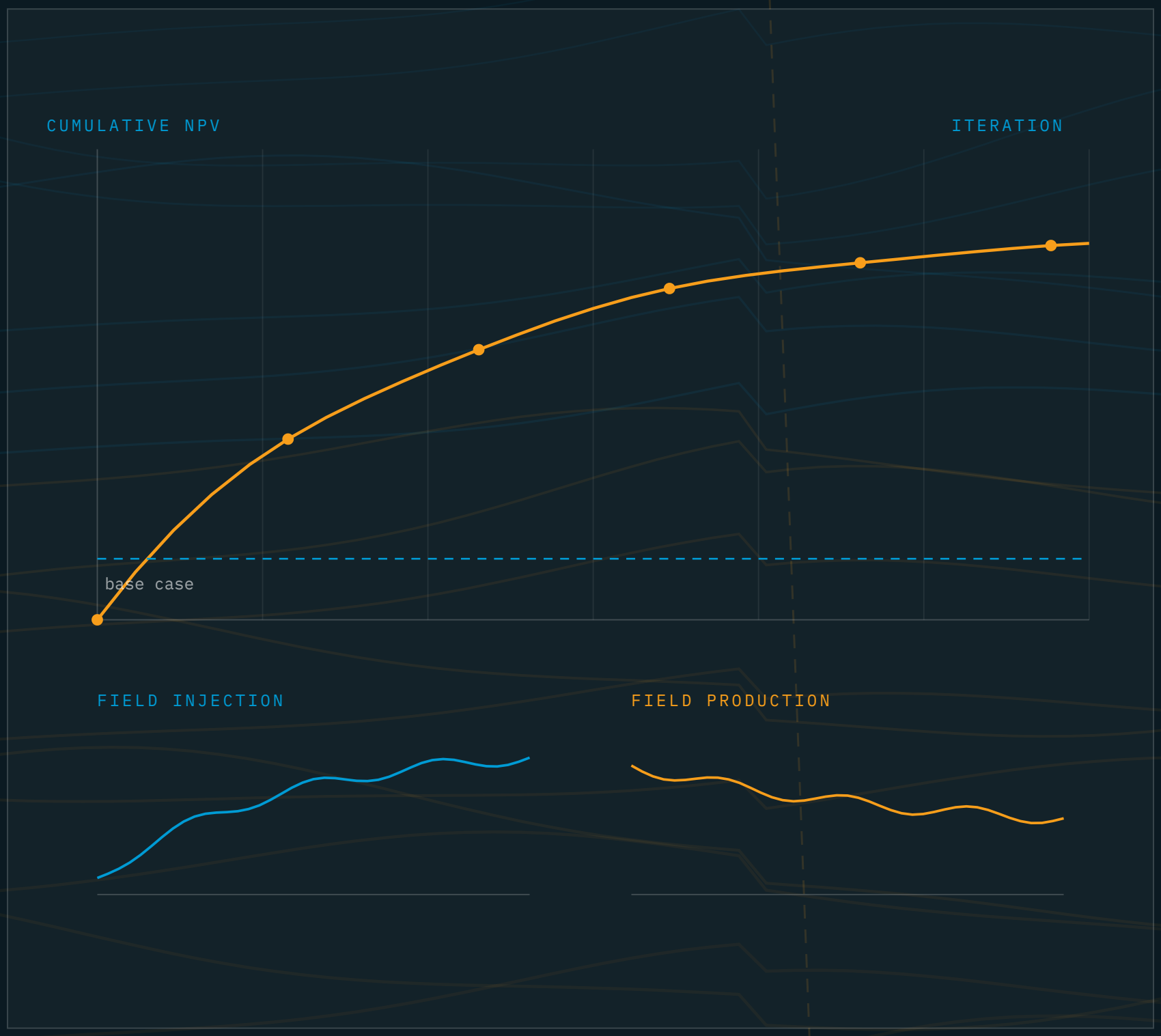


Fig. 04 · Schematic. Illustrative convergence shape, not project results.

RESULTS ACHIEVED

DEEPSOFT

What the integration delivered

Research method, operational tool

The short term constraint methodology developed at UnB became a tool integrated into Petrel, available to the reservoir engineers of the operating company that sponsored the research.

Architecture that survives

Embedding the Python interpreter instead of porting routines to C# keeps one codebase, so maintenance continues and new optimizers arrive without rework.

Native Petrel experience

Setup, execution, monitoring and analysis happen inside Petrel, with no manual exports and no drift between reservoir models and optimized strategies.

Advanced injection strategies

The integration supports techniques such as WAG, which matter for recovery factor in pre-salt carbonate reservoirs.

> Every case simulated along the way stays in the Petrel project, open to the rest of the platform.

Research software becomes an asset when it reaches the engineer's desk.

Storms Petrel was developed by DeepSoft as a software development service contracted by Finatec, connecting the STORMS optimizer built at UnB to the Petrel environment used by the operating company that sponsored the research. The resulting software belongs to UnB.

That is the work we do: engineering Petrel plug ins for teams that already own the science and need it to run where the decisions are made.

SERVICES

Petrel plug in development

CONTACT

contact@deepsoft.com.br

LINKEDIN

/company/deepsoft

STORMS and the resulting plug in are the property of Universidade de Brasília. DeepSoft acted as the software development contractor. Petrel is a trademark of its respective owner and its mention does not imply sponsorship or endorsement.

DEEPSOFT

Ocean Framework · C# · .NET · Python