

OCS WorkHub

Client: OCS Consulting (Internal Build)

Business Size: Internal Project

Industry: IT Consultancy / Professional Services

Country: UK

Technology: Angular 19, .NET 9, SQL Server, GitLab CI/CD, Claude Code

Objective: Consolidate time tracking, billing and support into one unified platform

The Brief

Prior to this project, OCS was managing timesheets, client billing and support ticketing across three separate systems, with limited integration between them. This meant that reconciling information across the systems was time-consuming and largely manual.

The objective was to replace these disconnected processes with a single, unified platform that brought time tracking, billing and support together in one place. The project also provided an opportunity for OCS to put AI-assisted development into practice, using the build as a live test of an approach that can now be applied to client projects.

The Challenge

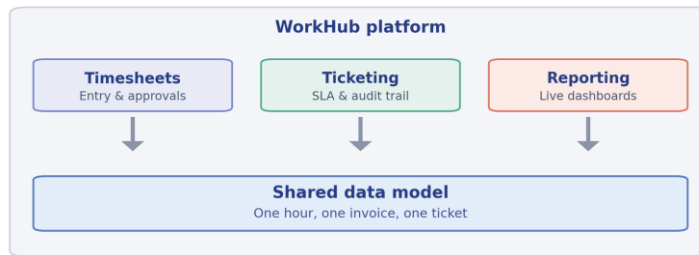
As OCS continues to grow, the existing systems support individual processes, but the lack of integration between them created opportunities to improve efficiency, visibility and consistency:

- Timesheets and billing sat in separate systems, meaning each invoice run required information to be reconciled manually between the two
- Support tickets had limited client and project context, making SLA tracking less consistent and difficult to rely on
- Reporting depended on manual spreadsheet exports, meaning management information could quickly become out of date
- There was no single view of consultant utilisation, making it harder to identify emerging capacity or resource issues early
- Approvals were managed largely through email, with no central audit trail to provide a clear record when timesheets or other information needed to be reviewed.

The opportunity was not simply to replace existing systems, but to build on what already worked and create a more connected, reliable and scalable way of working. The friction came from three systems that had evolved independently and were never designed to share data. Simply integrating them would have preserved many of the existing constraints, while an off-the-shelf platform would have meant adapting OCS processes to someone else's way of working.

The Solution

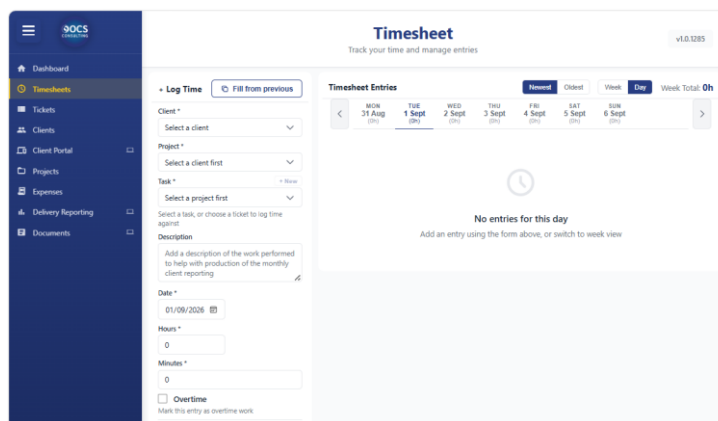
OCS built a single platform that brings time tracking, billing and support together around a shared data model. This means that information entered in one part of the system can be used across the relevant processes, reducing manual reconciliation and providing a more consistent view of the business.



The platform includes:

- **Timesheets** - time entry by client, project and task, with automatic overwork detection, multi-level approval routing and billing PDFs generated directly from approved hours
- **Ticketing** - a structured workflow with SLA tracking, a full audit trail of activity and templates for recurring service tasks
- **Reporting** - based on live timesheet and ticket data rather than overnight exports, therefore providing up-to-date management information when it is needed.

The result is a connected platform that builds on the existing OCS processes while providing greater consistency, visibility and control.

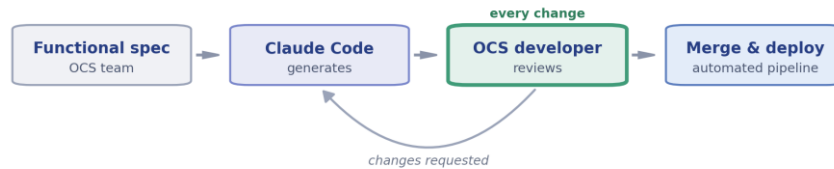


How We Built It

The platform was developed by the OCS team using Angular 19, .NET 9 and SQL Server, with GitLab CI/CD providing an automated build and deployment process. Claude Code was introduced as an AI-assisted development tool and used alongside OCS developers throughout the build.

AI was particularly useful for scaffolding, boilerplate, cross-layer consistency and elements of the automated test suite. It was also used to help review authentication flows and API design, providing an additional way to identify potential security gaps.

Every AI-generated change was reviewed by an OCS developer before being merged. The decisions about how the platform should work and how it should support the business remained with the OCS team.



Deployment is fully automated, with every change built, tested and released through a single pipeline. A database backup is taken before each release, and every version is traceable to the build that produced it. This has removed the need for manual deployment steps and provides a more consistent and controlled release process.

The Results

The new platform has delivered measurable improvements across the business:

20 hrs/month saved on timesheet-to-invoice reconciliation	14% reduction in helpdesk administration
30% improvement in on-time timesheet completion	3 → 1 systems consolidated into a single platform

In addition to these measurable improvements, the project has delivered:

- A single source of truth for time and service data, reducing the need to reconcile information across separate systems
- More consistent approval processes, with a complete audit trail replacing ad hoc email chains
- Live visibility of consultant utilisation and SLA compliance, allowing information to be viewed as it happens rather than retrospectively
- A practical example of AI-assisted development, giving OCS a working reference point for how AI can be incorporated into technology projects and applied to client work.

Lessons Learnt

- Start with the data model. The original challenges came largely from systems that had evolved independently and were not designed to share data. Establishing a common data model provided the foundation for a more connected platform
- Use AI where it adds the most value. AI-assisted development proved particularly effective for well-defined, repetitive tasks such as scaffolding, boilerplate, cross-layer consistency and elements of testing. It also provided useful support when reviewing areas such as security and API design
- Keep people at the centre of the process. AI helped increase the pace of development, but every generated change was reviewed by an OCS developer before it was merged. The technology supported the development team; it did not replace their judgement or knowledge of the business
- Use your own projects to prove what works. Building the platform within OCS provided an opportunity to test the approach in a real business environment, understand where AI-assisted development delivered the most value and identify how the experience could translate into future client projects.