



# Light Client

An ecosystem to streamline the development  
of new systems





# Introduction to Light Client

Why we decided to build such a platform? How our platform can speed up the process of building new software solutions? What software have we exactly created to full fill business requirements?

The background of the slide is a top-down photograph of a wooden desk. A person's hands are visible typing on a silver laptop. To the right is a white mouse and a white coffee cup on a saucer. In the bottom right corner, there are several documents. One document is titled 'GAS LOG' and another 'MAINTEN'.

DATE	AMOUNT	REMARKS
12.01.2013	14	
16.01.2013	17	
12.01.2013	14	

DATE	AMOUNT	REMARKS
09.01.2013	10	
18.01.2013	10	
01.01.2013	10	

# Let's start form **why**

## Why we decided to build such platform

While working on many programming projects, we noticed that very often a large part of the project's work, especially in its initial phase, is focused on creating basically the same components. The same applies to the frontend and backend parts.

Immediately after that, there is the problem of automatic testing and guaranteeing continuous integration and continuous delivery.

-  To limit the time that needs to be spent on pre-business implementations.
-  To have access to tested and ready to use components with 3<sup>rd</sup> party implementations

-  To have predefined and ready to use Layouts and styles that can be white-labeled
-  To have a library of UI modules that can be reused to limit time spent on creating UI



# Milestones of new projects

01

## “Standard” components

Let's start development from users (login, registration), then add authorization support, then audit logs. Same components for different projects

02

## Automated testing

We always want tests, but when time is pushing us forward, we always leave them for later. But this is an important part of a software project.

03

## 3<sup>rd</sup> party integrations

We might need to add support for providers like maps, email, SMS, payments, bank ID login, cryptography, etc.

04

## Demo environments

We will begin by establishing all build pipelines, database migration scripts, and Kubernetes clusters. Later, we will set up development, staging, and production environments.



# Software for **different types** of customers



## Costs of usage

Monolithic application vs. microservice-oriented? What about the platforms that this will be hosted on? How to Scale it to meet usage requirements



## Secure solution

Should we support data encryption in database? Do we need permission access model in our system? Should it be able to manually create new roles in the app?



## Customization

Same product, different customers. They want a different application layout. Apart from that, they want to have different content on the same pages.

“ Here is the simple but powerful rule... always give people more than they expect to get.

**Nelson Boswell**

# So **how** we did it?



## Ready to use Core Ecosystem

Created in the newest Dotnet bunch of microservices that when deployed will make fully functional APIs with most business-use components. Together with ready-to-use Cypress tests of them and support for CI/CD.



## Libraries with integrated solutions

We prepared libraries for PDF rendering, cryptography, email and sms sending with support of their customisation. Support of payment providers (eg. NETS, Przelewy24), and core Vue Npm UI Components

With those two parts, we can easily setup new project with deployment in the Kubernetes. A week after signing the agreement, the customer will get a ready-to-test environment and we can focus on implementing business logic.

# What are **three** key points?

01

## Architecture

Structure that supports different types of applications due to the microservices approach. Build with the most recent versions of dotnet and vuejs, then deploy to Kubernetes.

02

## Business

Components that speed up the process of building new solutions. The Admin Panel includes ready-to-use features such as whitelabeling, audit logs, users and permission management.

03

## Integrations

Ready to use integrations with payment providers, open street maps, SMS and E-mail. Modules for cryptography, PDF rendering and for UI development

# Key points of Architecture



## Dynamic content

Structure of pages, modules, roles and permissions, emails templates, layouts – everything configured in customer db.



## Cypress Tests

We cover our components by automated tests. We can easily extend them to new features because of built-in library.



## Customizable modules

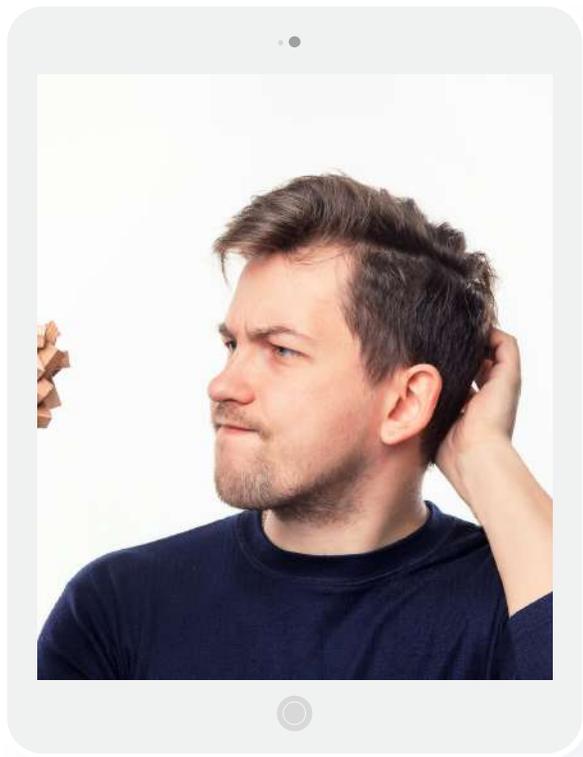
Modules have their own translations, permissions, and settings, allowing them to be tailored to each customer.



## CI/CD Automatization

Docker supports the entire ecosystem. Together with Kubernetes' reusable deployment scripts

# Business Components



## Multiple languages

Built in support for Polish, Norwegian and English languages. Easy to use component to manage translations



## Administration panel

Bootstrap approach layout with reusable components to speed up the process of adding new UI components.



## Users Component

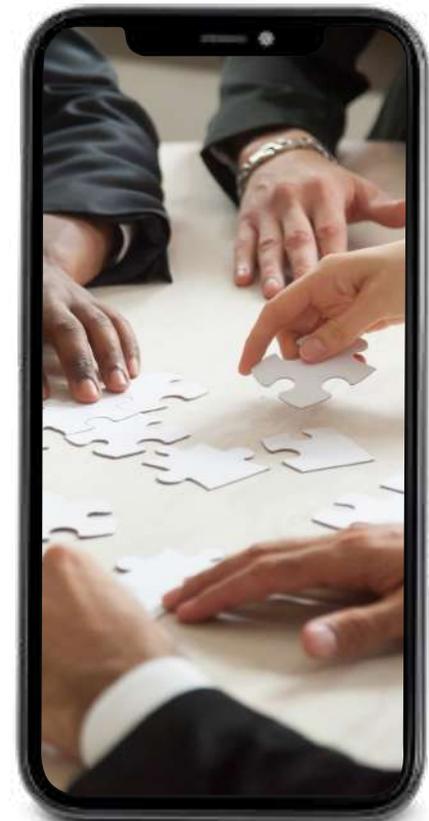
Ready-to use component manages users accounts with the possibility to assign roles, unlock and reset passwords.



## Audit logs Component

Built-in services to register and overview users' actions in the system.

# Ready to use Integrations



## Email & SMS

You can manage templates and sent it by providers like Microsoft 365 and Sendgrid



## White labeling

Possibility to brand colors for built in styles and UI components. Possibility to use custom styles with built it components



## Payment Providers

Payment providers for different countries, Nets, Przelewy24 or PayPal



## Manageable API Clients

Component for adding and managing API clients to support external sources accessing our API.

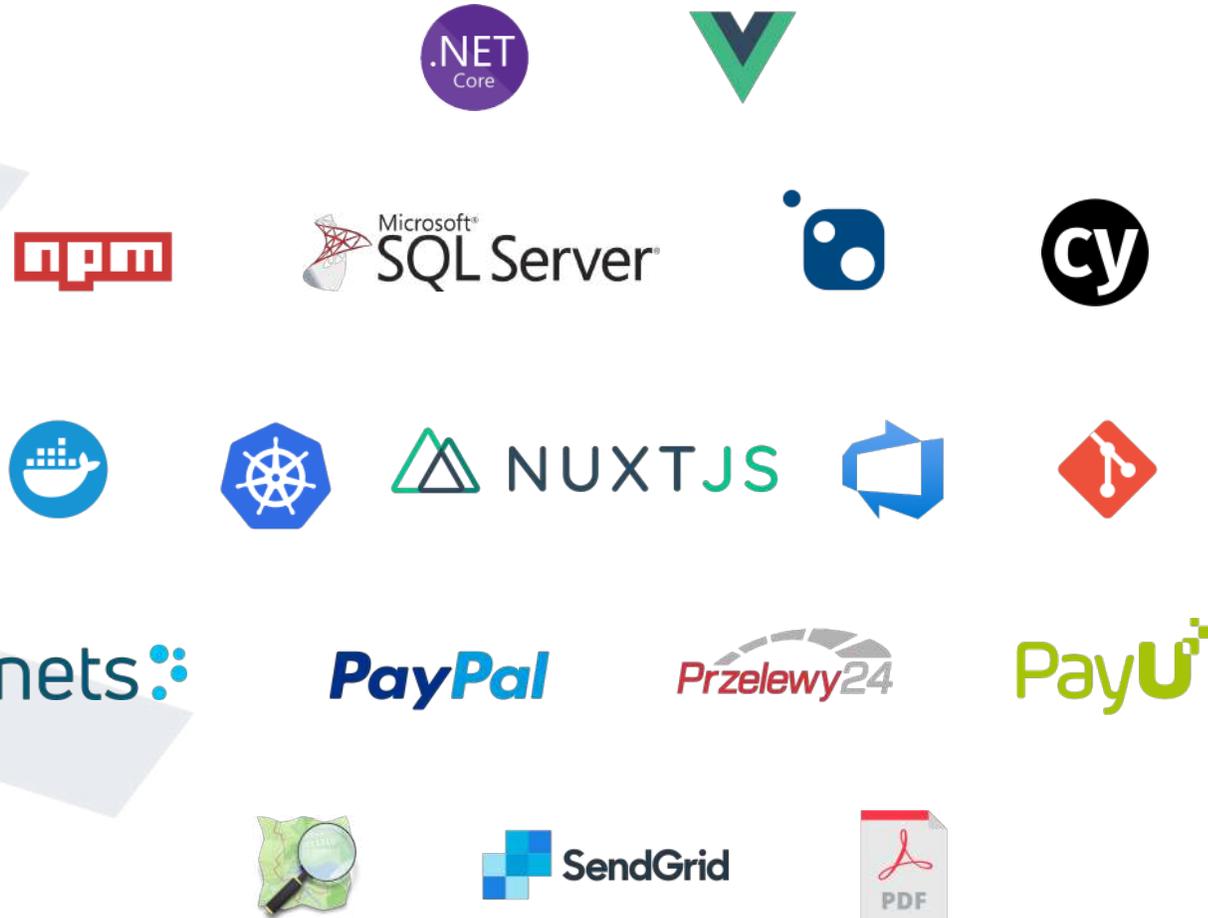


# So what really is Light Client

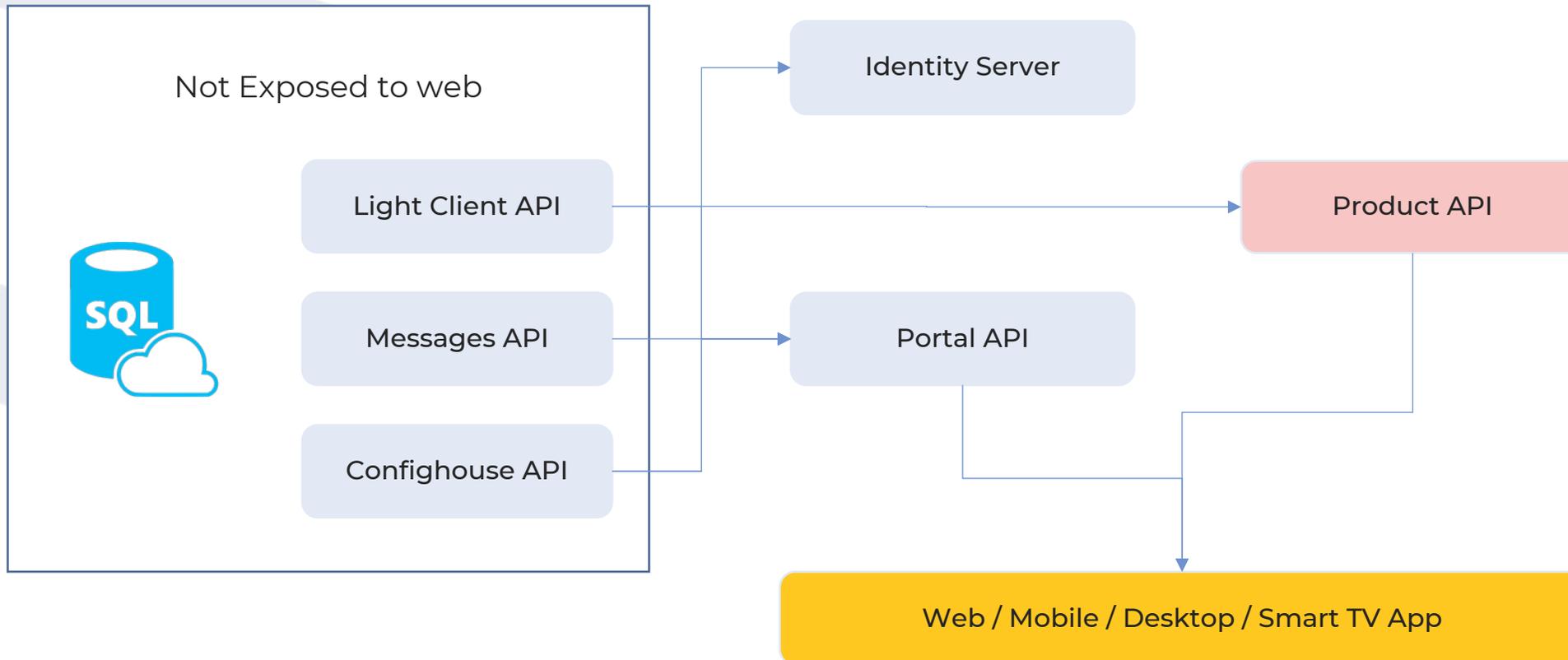
A brief demonstration of the solution, along with a description of the tools used to build this ecosystem



# What is **inside**?

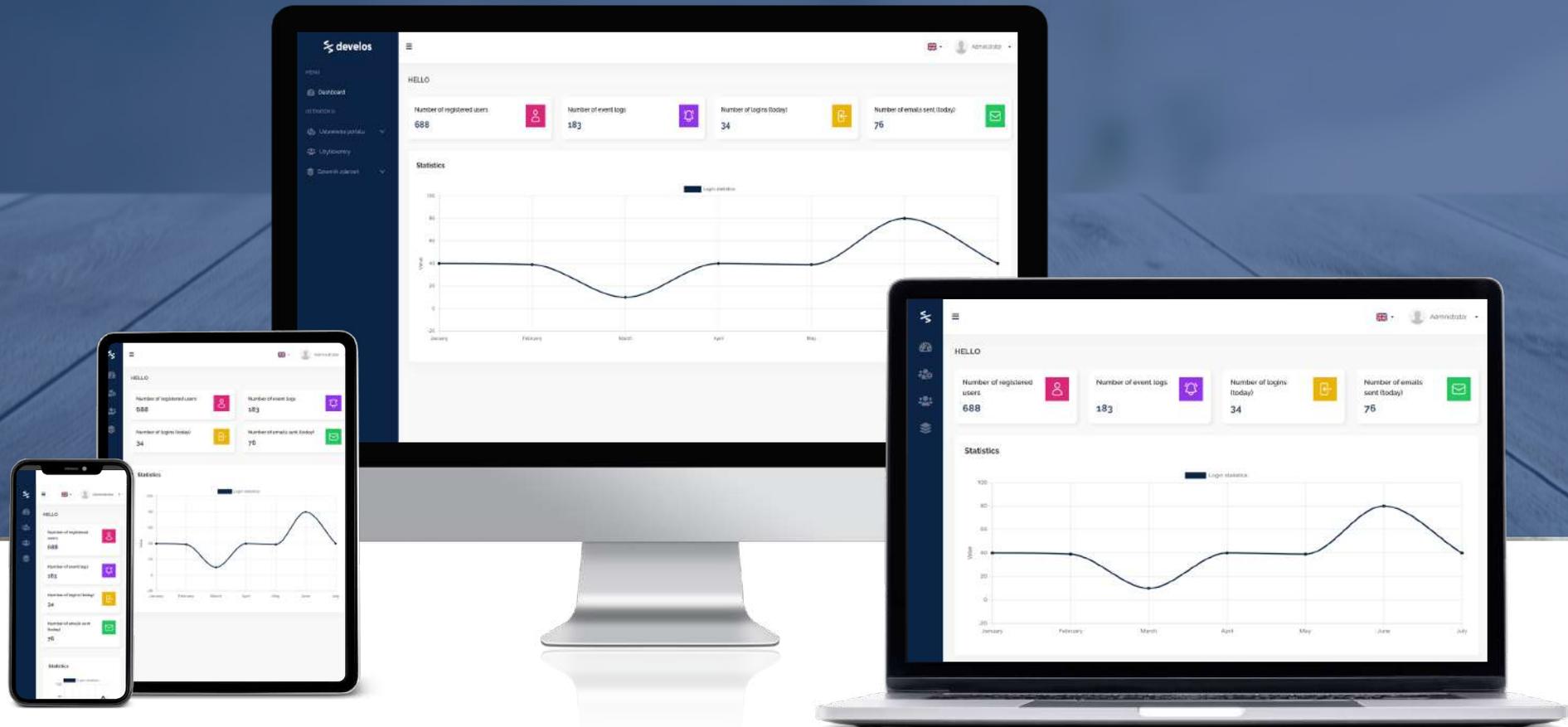


# Overview of architecture



# Light Client

As a standard nowadays, we know that all web applications must support responsive design. So this is something that Light Client supports out of the box because that was the goal for this. Out of the box, support as many reusable functions as possible.





**Thank you for  
your attention!**

---