

# Converged

## Introduction

Converged is a UI (User Interface) framework designed to create decentralized web applications, similar to how Android works for web platforms.

### Purpose

It's tailored specifically for technology startups, providing a seamless interface that can be used across both industrial systems and business applications.

### Purpose of Converged

- Controlling digital factories (managing thousands of devices)
- Direct control of industrial equipment (via web sockets)
- Managing a swarm of drones (warehouse robots, flying drones)
- Conducting scientific research
- Managing business processes

*The name "Converged" reflects its ability to handle various tasks common in high-tech businesses, symbolizing the merging of different functions and solutions.*

The framework features a modular architecture, making it flexible and easy to expand or modify. This means that anyone connected to the network can create and publish their own web module or web application. Furthermore, these creations are immediately accessible for use by any other network participant. Converged can support a vast array of visual interface modules, each with unique functionality.

## Differences

### Devices

### Computers

- Personal computers
- Laptops

For these options, a web interface with dynamically loaded modules is used.

## **Mobile Devices**

- Smartphones
- Tablets

For these options, a mobile application is used. The mobile application can be designed for both Android and iOS.

The mobile application employs the same dynamically loaded modules as the web interface, but they are adapted for mobile devices.

## **Modules**

Module loading happens dynamically from a decentralized platform. These modules can be cached on a local cluster or mobile device, eliminating the need for a constant internet connection.

Dynamic loading enables any user to work with thousands of available modules, using the latest versions.

## **Example**

Let's say you've purchased a 3D printer controlled by a Converged-based mobile app. All you need to do is install the Converged mobile app, connect to the printer via wifi, and you can manage it using the mobile app.

When using the mobile app, it automatically loads all the necessary modules for printer control and saves them. As new versions of modules appear, it updates them.

## **Connections**

One of the main distinctions of the Converged UI framework lies in the connection options it supports.

### **Connection Options to the Control Object**

- Direct connection to a decentralized platform, requiring no dedicated servers.
- Connection to a local cluster, for instance, a cluster of computers in a local network managing 3D printers.
- Direct connection to a device's microcontroller via the WebSocket protocol, e.g., connecting to a 3D printer.

### **Thus, there are 3 application scenarios for Converged:**

- Online usage (on the internet).
- In a local network.
- Direct interaction with the device (without the internet and without computers).

### **Compared to other projects**

In contrast to other open-source projects, which are typically aimed at managing 3D printers or CNC machines and are intended for single devices, Converged is focused on professional industrial and business applications.

### **Examples of such projects:**

- OctoPrint - a web platform for managing 3D printers, oriented towards home use and the control of individual 3D printers.
- Fluidd - a user interface for Klipper, which is also oriented towards home use and has the same limitations.

# **Advantages**

## **Advantages of Converged for Business Tasks**

- Thousands of ready-made software modules for solving various tasks.
- One-click installation of applications.
- Built-in mechanisms for data exchange between modules.
- Built-in management of business processes, such as order processing.
- Efficient utilization of cluster computing resources.

# Architecture

## Architecture

The Converged metaframework, based on Angular, draws inspiration from Vite.js and incorporates Angular and Bun.

### Differences from Basic Angular:

- Utilizes its own development system.
- Own build system.
- Own module loading system from the decentralized platform.
- Binary data exchange protocol SHOCK.
- Proprietary module caching system.
- Proprietary state management system.

**Micro Frontends** - These are interface modules that can be dynamically added and extended. Users can create and publish their own micro frontend on the Solenopsys platform, and other network participants can immediately use it.

# Principles

**Simplicity** - we focus on developing simple solutions. Frameworks have a straightforward architecture, and each individual module performs a simple function.

**Minimalism** - using a minimal set of fundamental elements in a variety of variations, with maximum reusability and universality.

**Reactive** - real-time interfaces, data streams, and event-driven architecture.

**Rejecting NGX**, if there is a high-quality component with well-written code, it can be integrated directly into UiMatrix, provided with a license, or linked to it in one of the ui-matrix libraries by creating a wrapper.

**Rejecting CSS frameworks**. For example, Material, Human, Fluent, Foundation, Bootstrap, Angular UI. These frameworks impose strict constraints on interface design and do not align with the task of creating real-time interfaces for digital factories, complex integration, and redundancy. We will use the best design system practices but focus on creating a monolithic, modern, fast, and high-quality real-time interface.