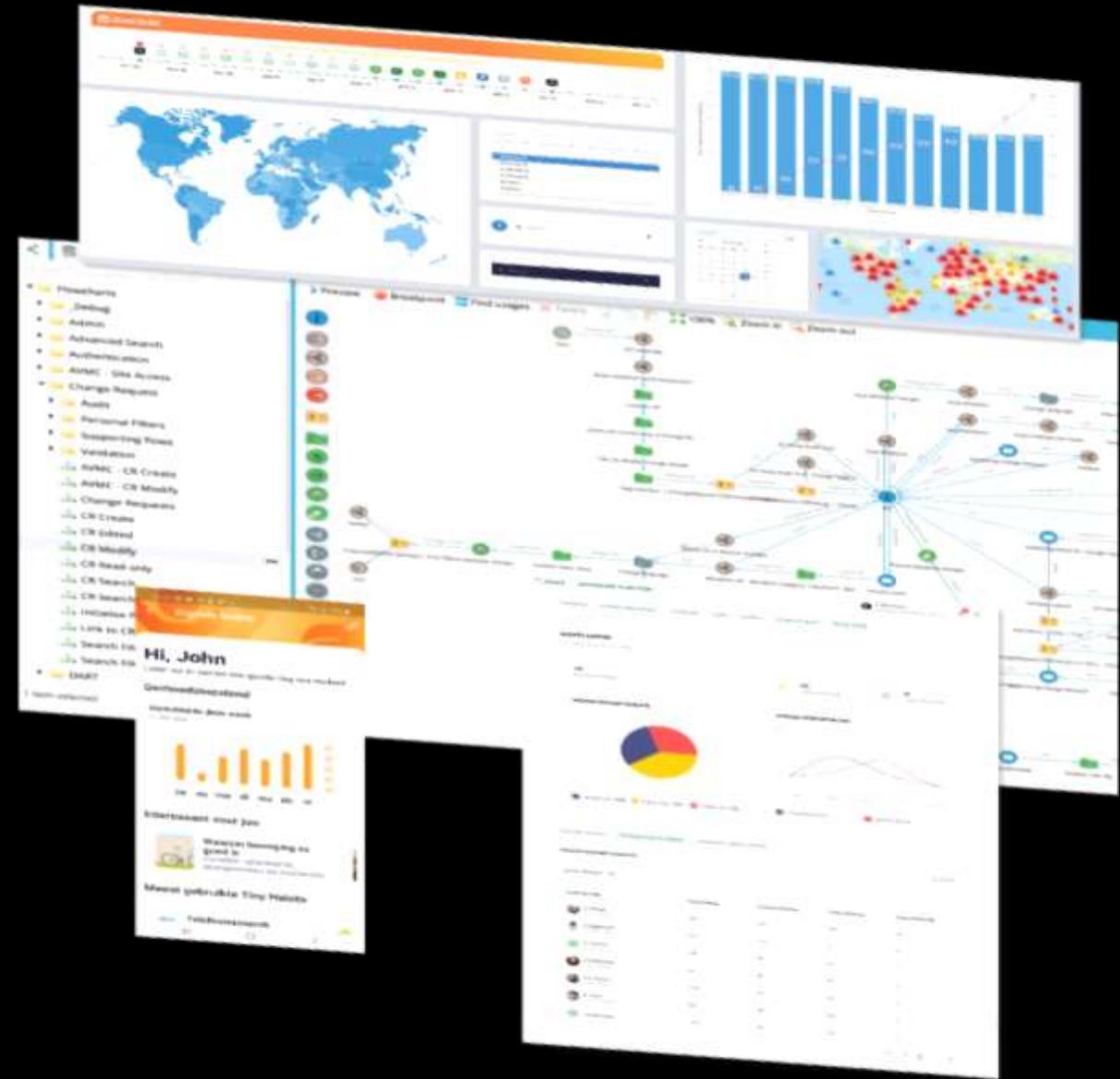




# S-Square - LowCode/NoCode (LC/NC) Enabling Technology Presentation

Jeff Friedman,  
VP, Sales & Customer Success

Version - 20221215\_V1



# Current Challenges in Traditional Application Development

## Long Development Timelines

- Custom development with standard SDLC processes
- Long incubation period before seeing a MVP
- Minor changes require long turn around time for design, build and testing.

## High Capital Expenditure and Operating Costs

- Investment in Software platforms and Infrastructure for custom development
- Higher support costs due to diverse support requirements

## Disparate Technology Landscape

- Multiple small projects using disparate technologies
- No uniform platform to manage small developments

## Developer Shortages

- Developer shortages and skill-set challenges
- Multiple small productivity projects get deprioritized

# 6 Generations of Programming Languages

First generation (1GL) - machine-level programming language used to program first-generation computers

Examples: machine-level programming languages

Second generation (2GL) - assembly languages. Examples: Assembly

Third generation (3GL) - more machine-independent (portable) and more abstract therefore more programmer-friendly than previous generations of languages

Examples: Fortran, COBOL, BASIC, Pascal, C, C++, Perl, Python, Java, JavaScript, Ruby, PHP, C#

Fourth generation (4GL) - include support for database management, report generation, mathematical optimization, GUI development, or web development. Examples: ABAP, Unix Shell, SQL, PL/SQL, Oracle Reports, R

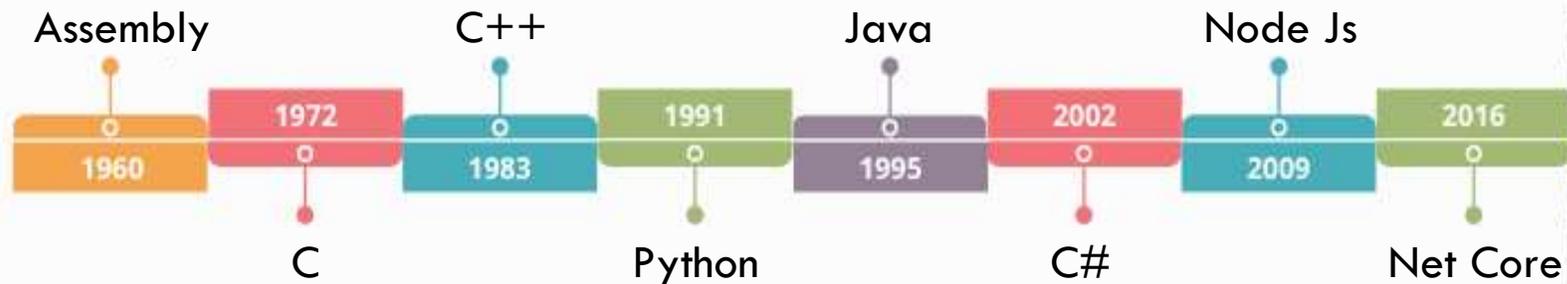
Fifth generation (5GL) - any programming language based on problem-solving using constraints given to the program to make the computer solve a given problem without the programmer, rather than using an algorithm written by a programmer. Examples: Prolog, OPS5, Mercury

Sixth generation (6GL) - programming language based on visual development. The overall umbrella term for these is "NoCode". Examples: Appian, WEM.io, Bubble.io

# Reinventing Software Development

## Traditional Coding

Requiring expensive, hard to retain code-linguists



Traditional computer languages require programmers to translate their thinking process into code built for the CPU and memory

## No-Code

Empowering transforming support to employ business-knowledgeable techno-functional resources



Optimized for how we humans think. Converting natural thinking process into working software

# Digital Transformation. Legacy Modernization. Business Velocity.

# 80%

## COST REDUCTION

Empowers employing business knowledgeable (techno-functional) resources instead of costly, hard to retain code-linguists to build, deploy and maintain secure scalable enterprise-grade software.

# 10%

## FASTER TIME-TO-MARKET

View app development in real-time. Deploy and update applications with a single click. Deliver software 10 times faster than traditional programming methods.

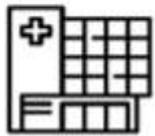
# 100%

## ALIGNED TO BUSINESS

Translate innovative business ideas to custom software built with no code app builder at the speed of, and fully aligned with, business requirements.



Banks,  
Financial  
Services and  
Insurance >



Healthcare >



Telecommunication  
>



Education &  
Training >



Manufacturing  
>



Public Sector  
>



Automotive  
>

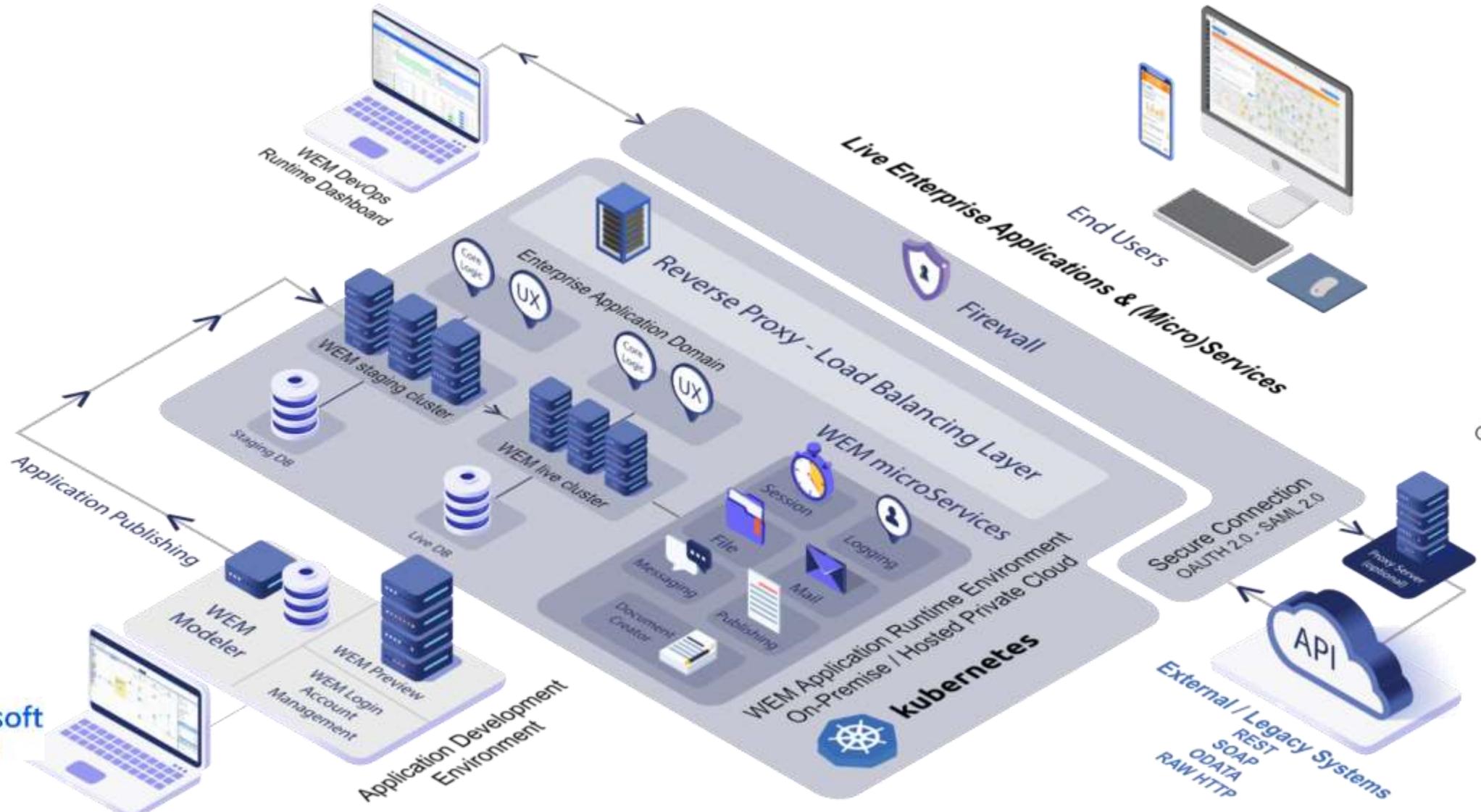


Real Estate  
>

# SCALABLE, SECURE CLOUD ARCHITECTURE



Google Cloud Platform



FLEXIBLE DEPLOYMENT OPTIONS FOR SHARED HOSTING, ON-PREMISE APPLIANCE AND PRIVATE APPLICATION CLOUD

# 3<sup>rd</sup> Party LCNC Marketplace Product Evaluation



Criteria	WEM	Betty Blocks	Power Apps	OutSystems	Mendix
Category	No Code	Low code	Low code	Medium to high code	Low code
Platforms	Web, native apps	Web apps	Web, native apps	Web, native apps	Web, native apps
Data Model	Drag & Drop	Visual Editor	Tables	Visual Editor	Visual editor
Visual Editor	Web-based	For backend apps	Web-based	Many designer	Web-based, desktop-based
Workflows	Drag & Drop	Action Modeler	MS Flow	Visual modeler	Visual modeler
Look & Feel	Custom templates	Custom js/css/html	Customizable	Custom js/css	Custom js/css
Environment	Public, private cloud, on premise	Public cloud, on premise	Public, private cloud, on premise	Public, private cloud, on premise	Public, private cloud, on premise
Release Management	Fully	Fully	Partially	Fully	Fully
Integration	All API standards	JSON, SOAP/REST	Office365, REST	SOAP/REST	SOAP/REST

# Use Case – Property Portfolio Management Application Modernization

Based in the United Kingdom, the company is a real estate market leader managing 6000 properties including office buildings, shopping malls, airports, logistics centers and 3,500 homes. The company is specialized in the technical management of buildings. The company provides effective property management following a one-stop-shop concept.

## PROBLEM

To be able to manage such a substantial number of square meters properly, the company required a good overview. The existing application was outdated and needed adjustments in its system, and the company was dependent on IT specialists for carrying it out. Many different parts are needed to be considered and checked regularly, this includes elevators, fire extinguishers, electricity, air conditioning and door handles and so on. These regular checks are essential to ensure the safety of the property itself, as well as of the people in and around it.

## SOLUTION

With WEM's no-code platform, the company developed a new application. This application provides an easy overview of all required checks and their current status. Moreover, it automatically sends an email, showing the relation of the property's status compared to security standards and legal inspections. It helps to accurately monitor these important checkups. It prevents those important elements that influence safety from being neglected.

### CUSTOMER CHALLENGES

- To manage data from a large number of properties.
- An easy overview of all required checks and their current status was required.
- The existing application was built on outdated programming language and was very complex.
- Adjustments in the existing system were complex to perform.

### WEM ADVANTAGES

- **Agile development, week to week results, short time to market (live in 3 months)**
- **Use of existing data from legacy systems/ integration with legacy systems**
- **Fast return on investment.**
- **Easy to extend the application**
- **New client portal for more self-service which resulted in increased customer satisfaction**
- **Cloud solution offers flexible workspaces (not tied to a location)**

# Representative WEM Enterprise Customers



# Thank You

Jeff Friedman,  
VP, Sales & Customer Success

## **S-Square Systems, Inc.**

4225 Executive Square Suite 600

La Jolla, CA 92037

+1 858-213-7063, +1 858-764-4441



**S-Square**

TRUSTED . TESTED . COMMITTED