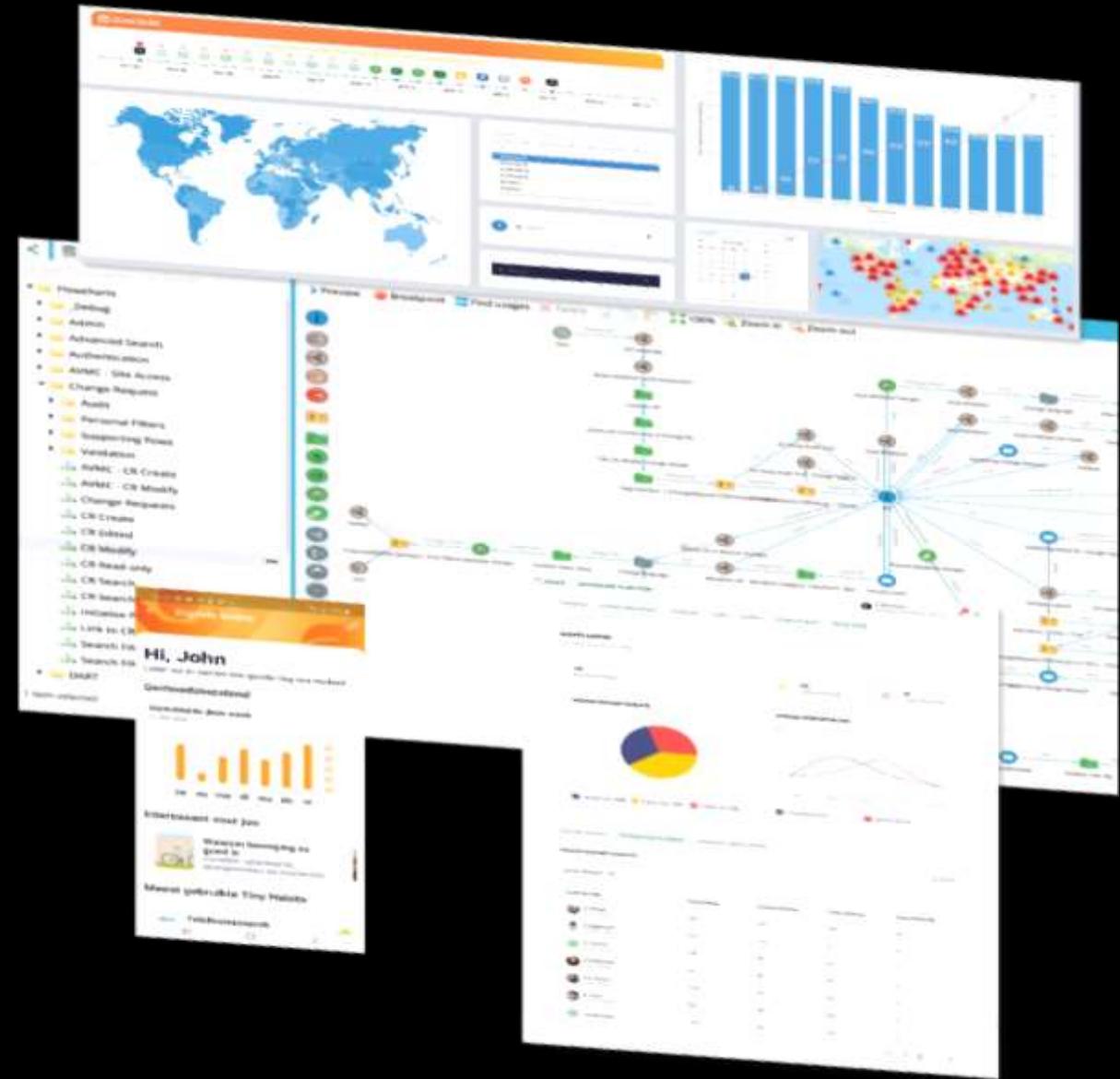




S-Square - LowCode/NoCode (LC/NC) Enabling Technology Presentation

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

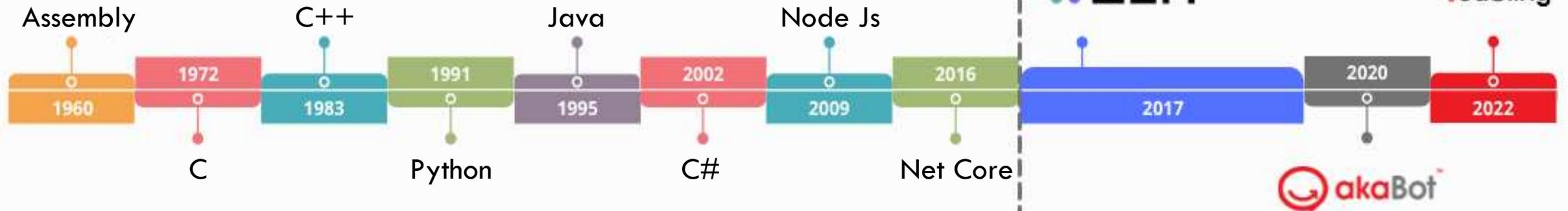
Reinventing Software Development

Traditional Coding

Requiring expensive, hard to retain code-linguists

No-Code

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



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

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.

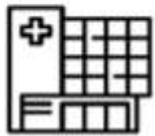
100%

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
>

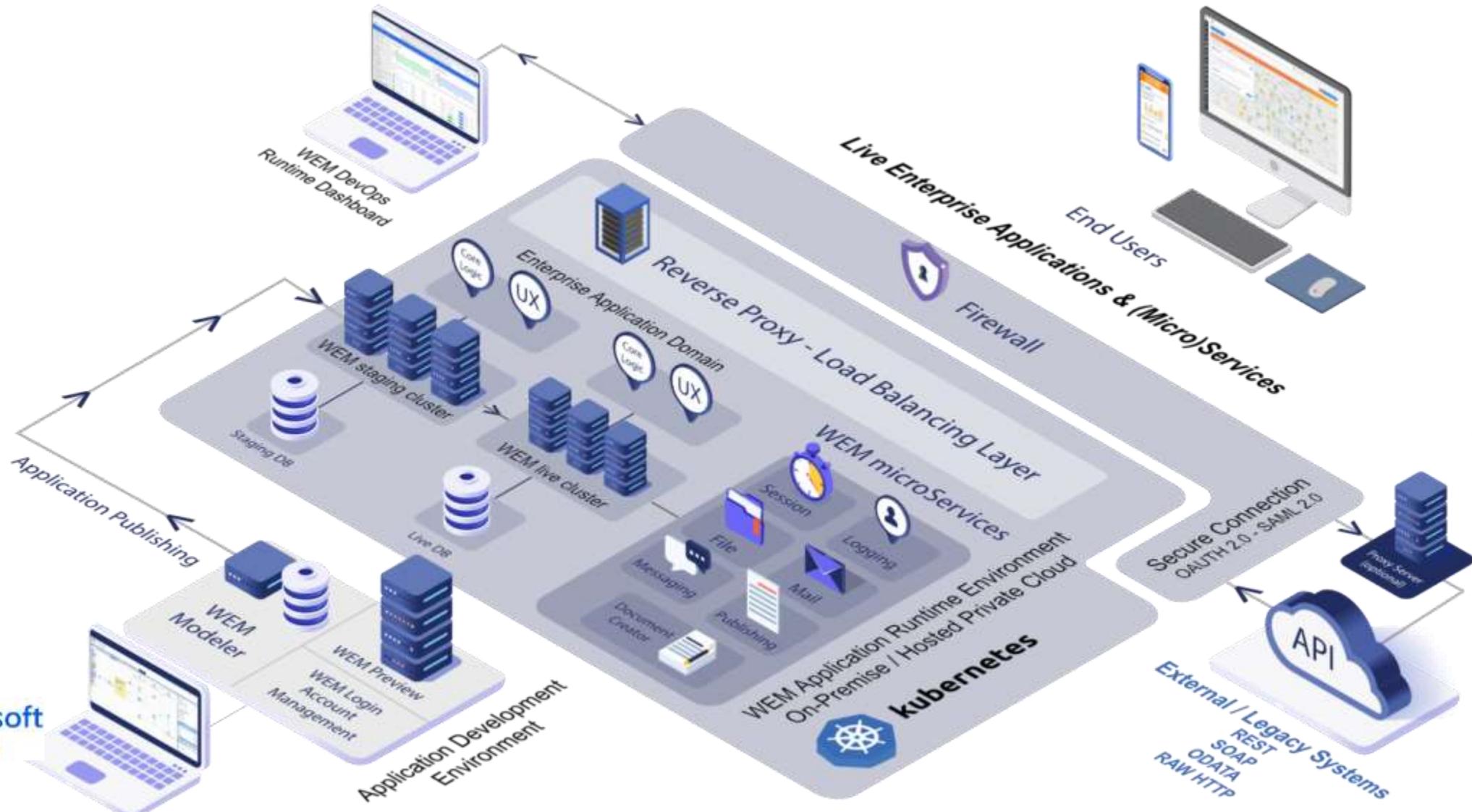
SCALABLE, SECURE CLOUD ARCHITECTURE



Google Cloud Platform



IBM Cloud



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

3rd 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 – Food Safety Certification

This Certification Organization has developed a food safety system certification which is the first worldwide standard for the entire food industry. The standard has been approved by the Global Food Safety Initiative (GFSI) and offers an opportunity to realize international harmonization. This certification provides a framework for effectively managing food safety, quality responsibilities and demonstrates that the company has a robust and effective food safety management system (FSMS) in place to meet the requirements of regulators, food business clients, and consumers.

PROBLEM

This Certification Organization used legacy systems, a large variety of Excel sheets, and several other administrative systems to support the food safety certification process. These systems had to be deployable worldwide (online) for certifying organizations with all the necessary security and backup functionality and that became a big challenge for the organization. It was not only supposed to be deployed online but also to maintain the source code and content. Because of the use of legacy systems and excel sheets the overall system was not web-enabled and not at all easily scalable. Reporting was one of the necessary features, but that was complex and labor-intensive.

SOLUTION

The organization had chosen to rebuild the entire application in WEM. By rebuilding, the new application would become future-proof, easy to maintain and expand, and fully backed up with the highest level of security. With the new application, adequate reporting tools were also immediately built-in. The organization had taken into account 1.5 years of development time to rebuild the new application but only after 2 months, the first part was live just because of the fast and agile development of WEM the organization could add more features faster into the new application.

CUSTOMER CHALLENGES

- To re-use the existing and historical data
- To change the process of certification frequently and to maintain and monitor it
- Integrate the new system with the existing legacy systems
- User interaction had to be very simple and easy to use for all users (skilled and unskilled)
- A two-way factor authentication was to be mandatorily developed.
- The system was supposed to have optimal performance (fast real-time response time)

WEM ADVANTAGES

- **From a complex and difficult to maintain legacy system and a large variety of Excel sheets to a clear, easy use and in the no-code environment, to be maintained by non-IT skilled employees**
- **With agile development, week to week results, short time to market the first deployment was live in 2 months**
- **Using existing data from legacy systems to offer a cloud solution, with flexible workspaces (not tied to a location)**
- **Create an easy-to-extend mobile application**

Representative WEM Enterprise Customers



JPMORGAN CHASE & CO.



MINISTERIO DE TRANSPORTES, MOVILIDAD Y AGENDA URBANA

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