Building multi-tier applications with Delphi

Pascal Conference
September 2025

Andrea Magni
andrea.magni@gmail.com

Andrea Magni

Monza, Italy - https://andreamagni.eu

Education: Computer Engineer
Artificial Intelligence and Robotics
Polytechnic of Milan

Job: Freelance / Trainer / Consultant

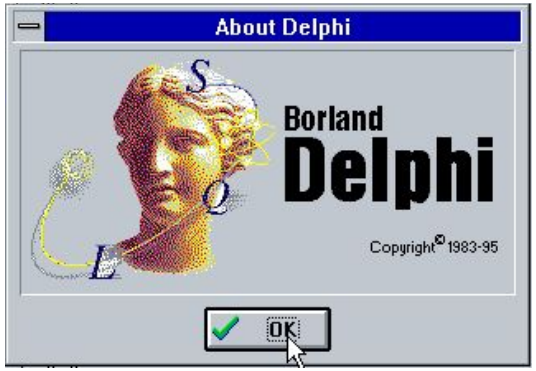
Platforms: Desktop, Server, Web, Mobile, IoT

Open Source: TFrameStand, TFormStand, FMXER
MARS-Curiosity REST library

Author: “Delphi GUI programming with FireMonkey”,
Packt Publishing, November 2020

embarcadero MVP Delphi

Delphi 1995 - 2025: 30 years



Windows 3.1 (16-bit)
Delphi 1
1995

Windows 95 (32-bit)
Delphi 2
1996

Windows XP (32-bit)
Delphi 7
2002

macOS (32-bit)
Delphi XE2
2011

Windows 10 (64-bit)
Delphi XE2
2011

iOS (32-bit)
Delphi XE4
2013

Android (32-bit)
Delphi XE5
2013

iOS (64-bit)
Delphi XE8
2015

Windows UWT
Delphi 10 Seattle
2015

Linux (64-bit)
Delphi 10.2 Tokyo
2017

macOS (64-bit)
Delphi 10.3 Rio
2018

Android (64-bit)
Delphi 10.3 Rio
2013

macOS Arm64
Delphi 11.0
2022

In the beginning…

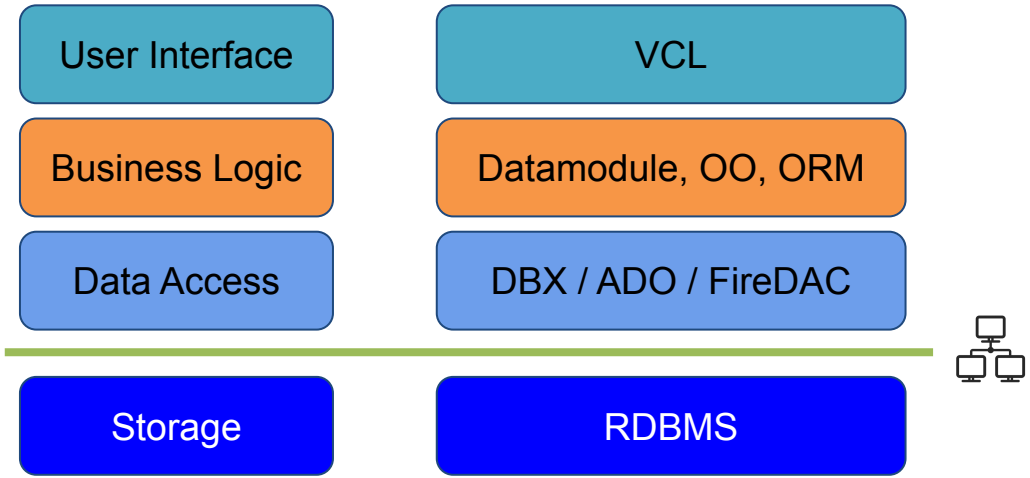
Data centric application

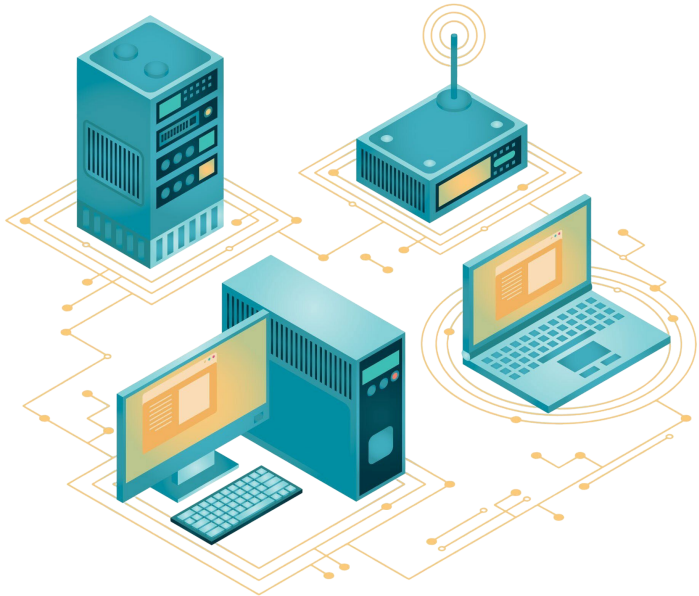
VCL
BDE
Local data

Architecture: local monolith

User Interface	VCL
Business Logic	Datamodule, OO, ORM
Data Access	BDE
Storage	Paradox, local DB, files

Architecture: client/server





Architectur lti-ti

Desktop app
Mobile app
Hybrids app
(TEdgeBrowser)

HTMX
Sencha
React
Vue
…

Flutter app
Mobile native app
SOA integrations

Frontend

VCL | FMX | Web | 3rd parties

REST Client | REST Client | REST Client | REST Client

Backend

REST Application server

DBX / ADO / FireDAC

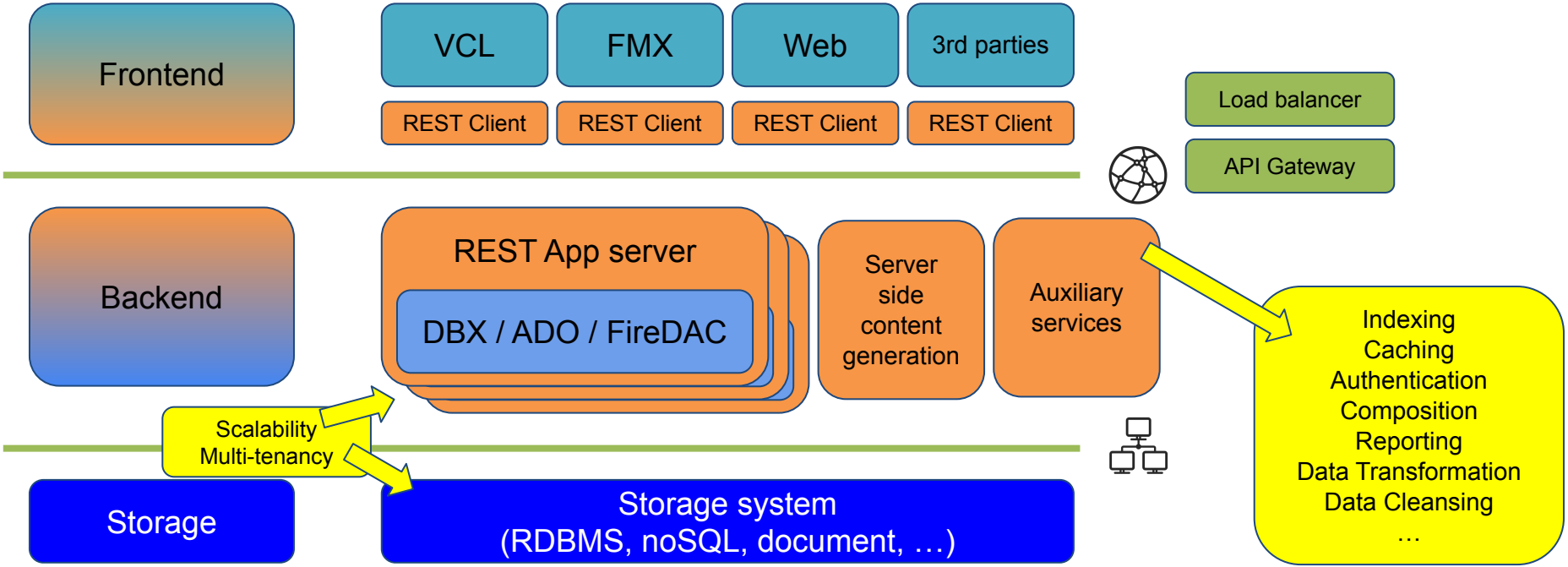
Server side content generation

Storage

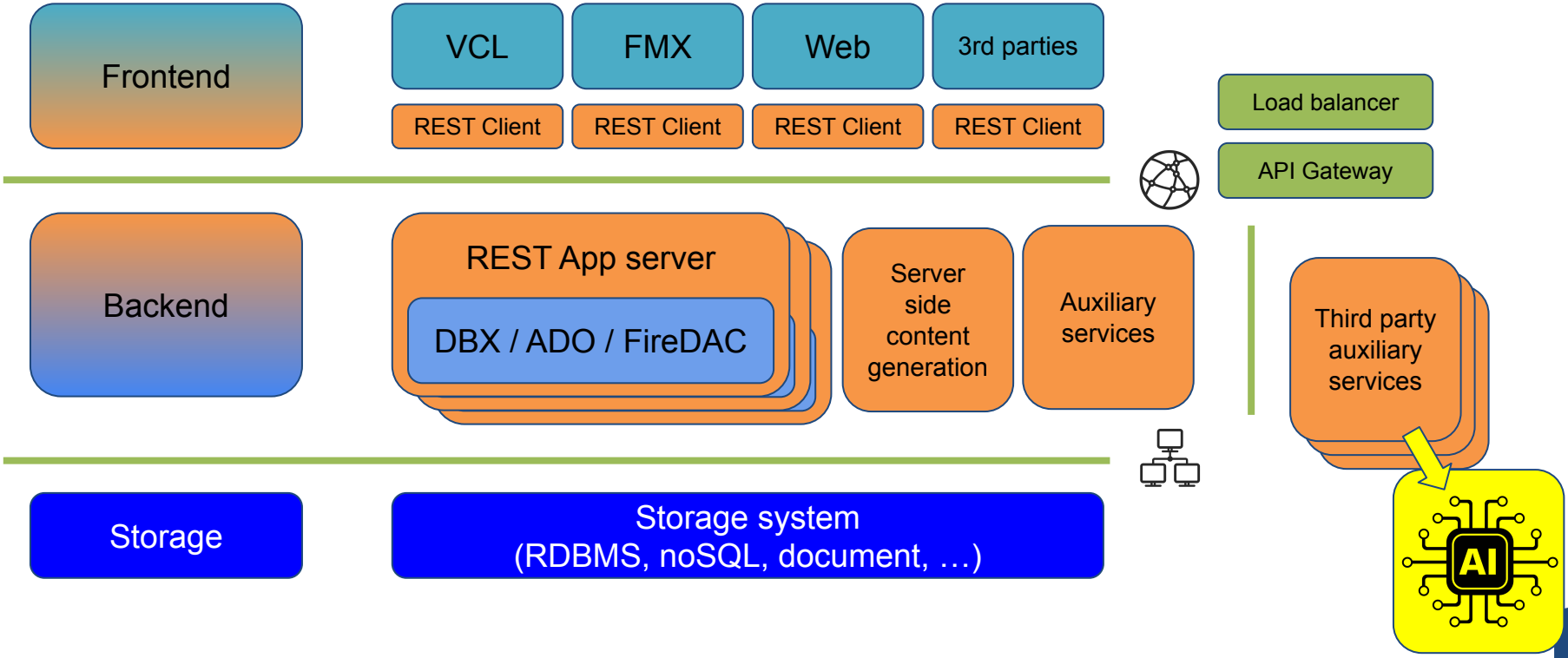
RDBMS

WebStencils
WebBroker
Templating
…

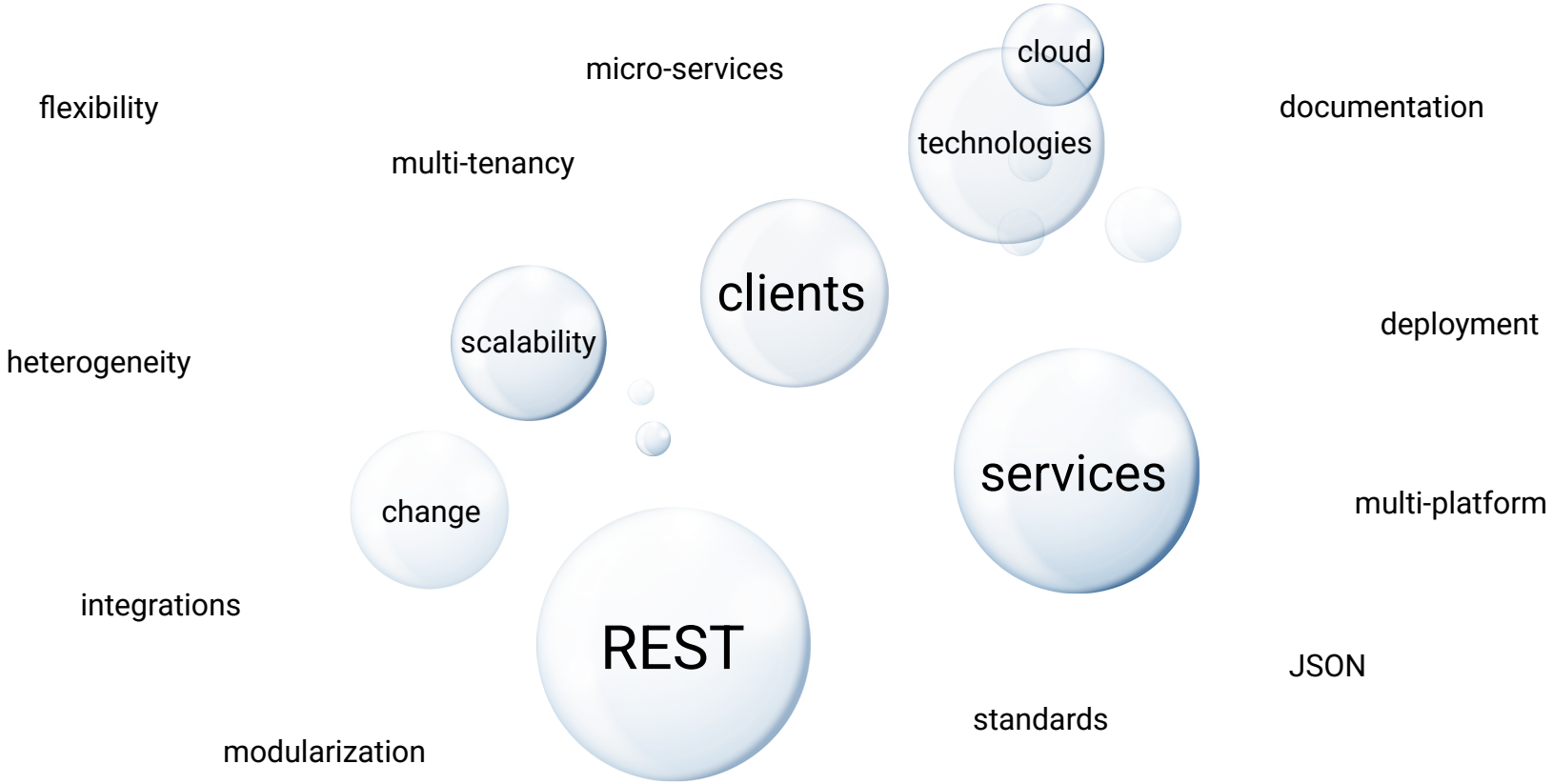
Architecture: multi-tier (2)



Architecture: today



A modern scenario



Why REST?

- REST (**Re**presentational **S**tate **T**ransfer) is important in modern software architectures because it provides a **simple**, **scalable**, and **widely adopted** way **to design APIs** that let **different** systems communicate over HTTP
- Key benefits:
  - **Interoperability**: standard HTTP methods (GET, POST, PUT, DELETE), accessible from almost any platform or language
  - **Scalability**: stateless, REST services can be easily scaled horizontally (with load balancers and distributed systems)
  - **Simplicity**: usually JSON or XML, lightweight (?)
  - **Flexibility**: Clients and servers evolve independently, as long as the API contract is respected
  - **Ubiquity**: REST has become the *de facto* standard for web APIs, supported by almost every modern framework and tool
- SOAP, CORBA, proprietary protocols < REST < GraphQL, gRPC, …

REST with Delphi

Official	Commercial	Open Source	OpenSource	OpenSource	OpenSource
RAD SERVER	**XData**	**DMVC Framework**	**mORMot 2**	**Horse**	**MARS**
Embarcadero	TMS Software	Daniele Teti	Arnaud Bouchez (Synopse)	HashLoad	Andrea Magni

Other options:
WebBroker/Datasnap

Why MARS?

MARS-Curiosity REST Library

Disclaimer: author *may* be biased!

- **Standard**
  - Web (HTMX, Sencha, React, jQuery, …)
  - other languages/technologies (Flutter, JS, Java, C#, Python, php, …)
- **Delphi-like**
  - modern language features (Attributes, DI, anonymous, generics, ...)
  - Delphi to Delphi functionalities (i.e. FireDAC integration, …)
- **RAD support**
  - client library and design time support!
- **Lightweight library** (no architectural imposition)
  - plug-in whatever you need/use
  - DAC library, ORM, JSON library, …
- **OpenSource:** https://github.com/andrea-magni/MARS

Demo time

- Installer
- MARSTemplate
- MARSCmd
- JWT Authentication/Authorization
- Serialization/Deserialization
- FireDAC integration
- OpenAPI3
- Client library