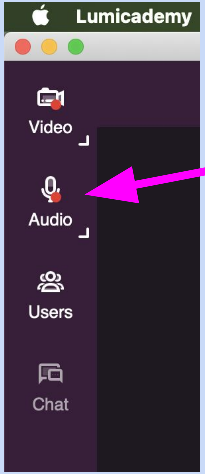
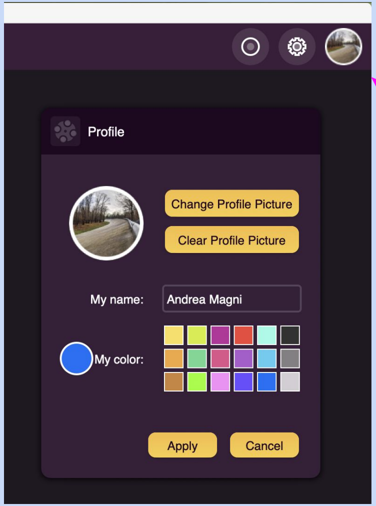
- La piattaforma permette ai partecipanti di condividere la vostra webcam e il microfono
  - Per cortesia, aprite il microfono solo per fare domande
- Per accedere alla chat Slack:
  - https://andreamagni.eu/live.html



Pallino rosso = spento, click per attivare (vale per Video e Audio)



My name: Andrea Magni

My color: ●

Potete impostare il vostro nome utente, se volete

Delphi Live - Italia
08 Aprile 2020

Andrea Magni
me@andreamagni.eu

Andrea Magni

Ingegnere informatico, Monza

Sviluppatore, Formatore, Consulente

Desktop, Server, Web, Mobile e IoT

TFrameStand, TFormStand, FMXER (FMX)

MARS-Curiosity (REST Library)

FMX Book (coming)

embarcadero MVP Delphi

https://andreamagni.eu

Piattaforma Lumicademy

- https://www.lumicademy.com/
- https://blog.grijjy.com/
  - Allen Drennan
  - Erik van Bilsen
- Applicazioni FMX
  - Windows e OS X
  - con integrazioni a basso livello (GPU, video)

Agenda

- MARS-Curiosity
- Autenticazione e Autorizzazione
- Accesso ai dati
- Deploy
- Client library
- Q&A e saluti

Why REST

REpresentational **S**tate **T**ransfer is a set of constraints that ensure a scalable, fault-tolerant and easily extendible system.

- Decoupling
- Good architectural style for **distributed** systems. Typical scenario: intermittent **network** / huge number of **heterogeneous** clients (web, mobile, desktop, SOA)
- http + JSON (or XML)
- Roy Fielding, 2000

REST with Delphi

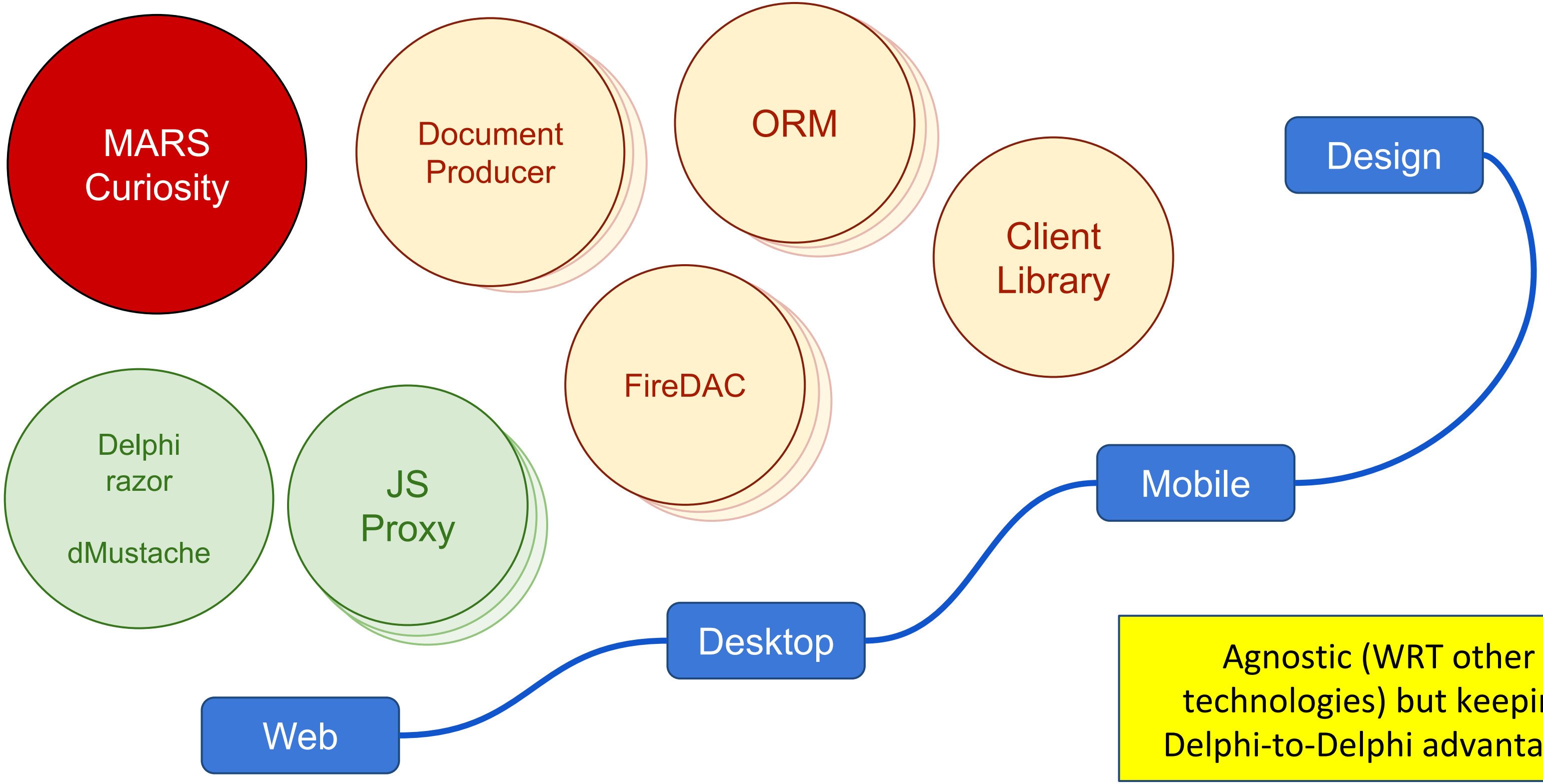
- **Datasnap** (Embarcadero)
- **RAD Server** (Embarcadero)
- **XData** (tmssoftware.com)
- **DMVCFramework** (Daniele Teti)
- **mORMot Framework** (Synopse, Arnaud Bouchez)
- (others)
- **MARS-Curiosity** (me)

MARS-Curiosity

Disclaimer: author *may* be biased!

- **Standard**
  - Web (jQuery, AngularJS, ExtJS, …)
  - other langs (Java, C#, 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**

The big picture



MARS Curiosity

Document Producer

ORM

Client Library

FireDAC

Delphi razor

dMustache

JS Proxy

Design

Mobile

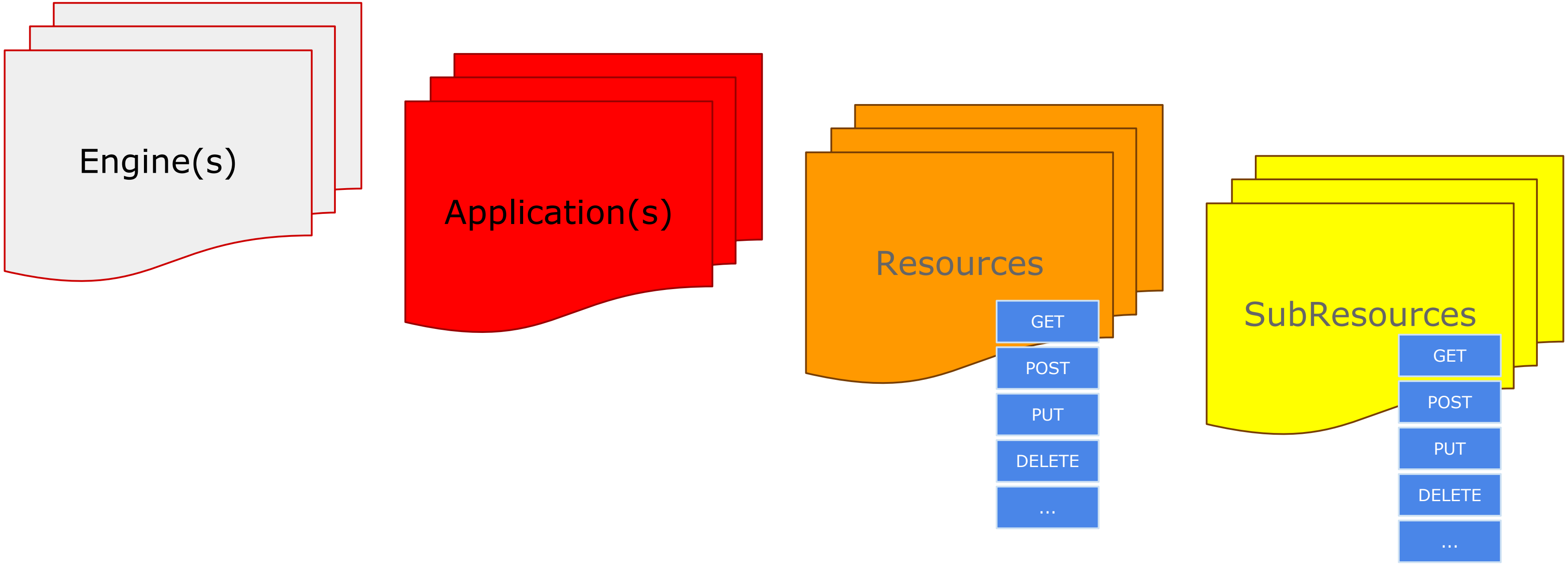
Desktop

Web

Agnostic (WRT other technologies) but keeping Delphi-to-Delphi advantages

Basics

```
http://localhost:8080/rest/default/helloworld/subresource/123?filter=best
[Engine Base Path]/[application]/[resource]/[subresource]/[PathParam]?[QueryParam=Value]
```



Automatic (de)serialization

- **MessageBodyWriter** mechanism

  Delphi datatypes (even custom ones) as function’s result

  Serialized with respect to client’s request.

- **MessageBodyReader** mechanism

  Delphi datatypes (even custom ones) as arguments

  Deserialized from client’s request content (body)

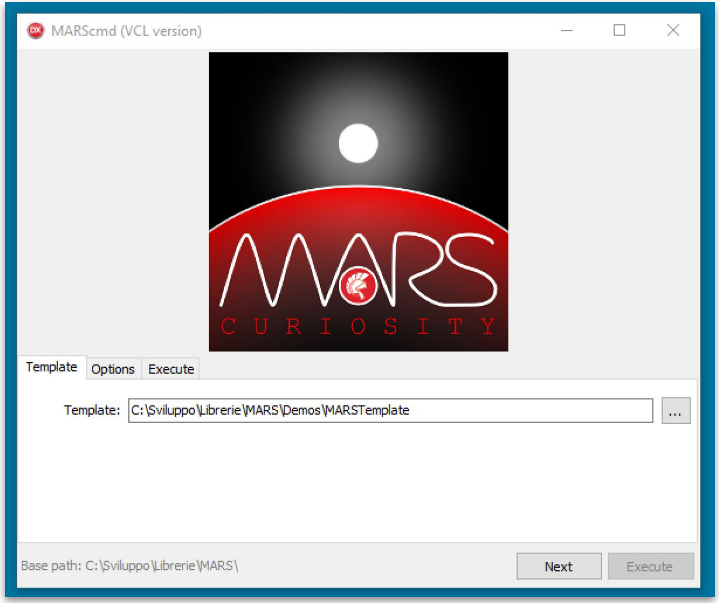
Get started

- Get a copy of MARS (git clone or download zip)
  - https://github.com/andrea-magni/MARS
- Add three folders to your Library Path:
  - `\Source`
  - `\ThirdParty\delphi-jose-jwt\Source`
  - `\ThirdParty\mORMot\Source`
- Packages (example for 10.3 Rio):
  - Open `\Packages\103Rio\MARS.groupproj,` Build All
  - Open `\Packages\103Rio\MARSClient.groupproj,` Build All
  - Install `MARSClient.CoreDesign`
  - Install `MARSClient.FireDACDesign`

An easy way to support this project is to give a "star" on GitHub!

MARSCmd

- Jump start a new project
- Cloning `Demos\MARSTemplate` project group
  - VCL App
  - FMX App
  - Console App
  - Windows Service
  - Apache Module
  - IIS ISAPI
  - Linux Daemon
- Shared initialization and resources
  - Server.Ignition.pas
  - Server.Resources.pas



Template: C:\Sviluppo\Librerie\MARS\Demos\MARSTemplate

Base path: C:\Sviluppo\Librerie\MARS\

JavaScript Web Tokens: JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM
0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV
9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

https://jwt.io/introduction/

https://auth0.com/learn/json-web-tokens/

Key aspects:

- ~~Standard~~ Quite popular
- No server-side session data ➔ Scalability

Role-based authorization

```
type
  [Path('item'), Produces(TMediaType.APPLICATION_JSON)]
  TItemResource = class
  private
  protected
  public
    [GET, PermitAll]
    function Retrieve: TJSONObject;

    [PUT, RolesAllowed('poweruser', 'admin')]
    function Update([BodyParam] AJSONObj: TJSONObject): TJSONObject;

    [POST, RolesAllowed('admin')]
    function Add([BodyParam] AJSONObj: TJSONObject): TJSONObject;
  end;
```

Features 1/2

- Parameters: (Path|Query|Header|Cookie|Body)Param
- MBR / MBW
    - multiple Consumes/Produces
    - custom Consumes/Produces
- Record/JSON functionalities
- Object/JSON functionalities
- DataSet/Record functionalities
- JWT Authentication and Authorization
- RBAC
- FireDAC integration
    - code
    - design-time

Features 2/2

- Client library
    - Async functionalities
- Metadata
    - Documentation
    - Swagger
- Invocation hooks (Before|After)Invoke + Error handlers
    - Logging

DelphiRazor

- https://github.com/marcocantu/DelphiRazor
- Demos\DelphiRazor
- Demos\Gallery

- Server side processor
  - HTML but not only
- TODO: contribute to DelphiRazor (more collections)
- Compiles for Linux platform

dMustache

- https://github.com/synopse/dmustache
- Demos\Mustache

- Server side templating
  - Logicless
  - HTML but not only
- Not supported on ARC enabled compilers (mobile, Linux)

FireDAC integration

- ~~Uses FireDACJSONReflect~~
    - Brand new implementation in v.1.2 !
- ApplyUpdates (delta-like politics)
- Design time support (server and client-side)
- But also expose FireDAC dataset’s as standard JSON
- MediaType (Accept): **application/json**
    - [{ “field1”: “value1”, “field2”: “value2”, …}, { … }]
- MediaType (Accept): **application/json-firedac**
    - { “dataset1”: “eeCBSdaFDSFDXX...”, “dataset2”: “sddkaffDSDfG….” }

FireDAC (DI) - Injection Service

- MARS.Data.FireDAC.InjectionService.pas (TMARSFireDACInjectionService)
- Injection provider for
    - TFDConnection (+TFDQuery?)
        - ConnectionDefName
            - ConnectionAttribute
                - Field, Property, Param annotation
                - Method annotation
                - Resource class annotation
            - Fallback to a default connection def name (config) <AppName>.FireDAC.ConnectionDefName=TODO
    - TMARSFireDAC

FireDAC (DI) - TMARSFireDAC (1)

- Helper functions to create FD instances (preconfigured)
  - CreateCommand / CreateQuery
  - CreateTransaction
  - Connection property
  - ConnectionDefName property
- Helper functions/procedures to perform common operations:
  - Query
  - ExecuteSQL
  - InTransaction
- Connection definitions helper functions:
  - LoadConnectionDefs
  - CloseConnectionDefs
  - CreateConnectionByDefName

FireDAC (DI) - TMARSFireDAC (2)

- Macro and Parameter’s value injection
  - Token context
    - Username and all TMARSToken properties
    - HasRole (Token_HasRole_admin)
    - Claim (Token_Claim_ACCOUNT_ID)
  - Request context
    - PathParam / QueryParam
    - Request fields (TWebRequest properties)
    - URL fields (TMARSURL properties)
    - URL prototype fields (TMARSURL properties)
  - Custom providers (define your own)

Client library

- Set of classes
- Set of components (non-visual)
- Design time support (request execution and metadata)
- Standard + Delphi specific functionalities (i.e. FireDAC)
- Error handling
- Multiple async executions
- Record serialization/deserialization

Web applications

- Every (modern) web technology has a way to get data from/put data to a REST server
- jQuery ($.ajax)
- AngularJS ($http service)
- ReactJS
- XHR request (raw)
- Server side page scripting
- Sencha ExtJS
- …

What’s next

- Documentation?
- Use Metadata within server internals (performance boost?)
- Use Metadata to drive IDE functionalities (component editors)
- Improve client library
  - (De)serialization mechanisms on the client-side
- Emit proxies?
  - Javascript (jQuery, AngularJS, ReactJS…)
  - Delphi (and other languages?)

Resources

Dedicated sub-forum at Delphi-Praxis EN:

https://en.delphipraxis.net/forum/34-mars-curiosity-rest-library/

Blog posts tagged mars-curiosity on my blog

http://www.andreamagni.eu/wp/tags/mars-curiosity

GitHub’s Wiki

https://github.com/andrea-magni/MARS/wiki

Tutorials and webinars:

https://github.com/andrea-magni/MARS/wiki/Resources

Grazie