MULTITIER → CLOUD

Best Practices & Strategies

wintech italia

Paolo Rossi

Wintech Italia **CTO**

Delphi **Dev**

Web Dev

DelphiDay italian conference

PADOVA EDITION

Edizione 2019

wintech italia

PAOLO ROSSI

SENCHA
MVP

EMBARCADERO
MVP

wintech italia

BLOG

CODING IS FUN

blog.paolorossi.net

DelphiDay
italian conference
PADOVA EDITION
Edizione 2019

wintech italia

GITHUB
PROJECTS

github.com/paolo-rossi

DelphiDay
italian conference
PADOVA EDITION
Edizione 2019

wintech italia

GITHUB PROJECTS

Delphi JWT

JSON Web Token Library for REST
(and not only REST)

Delphi Neon

JSON Serialization Library for REST
(and not only REST)

PADOVA EDITION

DelphiDay italian conference

Edizione 2019

wintech italia

GITHUB PROJECTS

Linux Daemon

Library to build real Linux daemons

OpenAPI-Delphi

OpenAPI 3.0 library for Delphi

DelphiDay italian conference

PADOVA EDITION

Edizione 2019

wintech italia

GITHUB PROJECTS

WiRL Project

JAX-RS Like REST Library for Delphi

DelphiDay italian conference

PADOVA EDITION

Edizione 2019

wintech italia

GITHUB PROJECTS

github.com/paolo-rossi

DelphiDay italian conference

PADOVA EDITION

Edizione 2019

wintech italia

AGENDA

- Microservices, a recap
- The cloud, what is it?
- Moving the architecture to the cloud
  - What, When and Why
- Cloud providers
  - “Locals” vs “the big names”
- IT infrastructure
  - As usual: Linux vs Windows
  - Docker or not Docker?

MICROSERVICES

N-TIER ARCHITECTURE

- Web apps
- Mobile apps
- DB-based apps (classic C/S apps)
- Automation apps
- IoT gateway/supervisor

MICROSERVICES

- ➔ Independent Deployability
- ➔ Modeled Around a Business Domain
- ➔ Own Their Own Data

ADVANTAGES

- ➔ Can be worked on in parallel
  - ◆ Scale the number of developers
  - ◆ Easier to understand their part of the system
  - ◆ Easier to get external help in a (small) part of the architecture
- ➔ Process isolation
  - ◆ Possible to use different technologies (mixing)
  - ◆ Possible to use different programming languages
  - ◆ Possible to use different deployment platforms
  - ◆ Possible to use different databases

ADVANTAGES

- Flexibility
  - More options regarding how you can solve problems in the future
- Improve robustness
  - Functionality is decomposed
  - A problem in one area does not bring down the whole system
- Embrace new technologies
  - Monoliths limit our technology choices
  - We have one programming language on the backend
  - We're fixed to one deployment platform, one operating system

PROBLEMS

- Main problem: networks
  - Communication between computers over networks is not “real-time”
  - Latencies
  - Network failure
  - Activities that are relatively simple with a single-process monolith now become problematics
- Consistent view of data across multiple machines

SUMMARY

Microservices “buy you options”

- ➔ Microservices are great but..
- ➔ They have a cost
- ➔ You have to decide if the cost is worth the problems

THE “CLOUD”

THE PROBLEM

You have the data, the services, the apps

- ➔ Now: how to reach these services from the outside?
- ➔ You have to move the data to “the cloud”
- ➔ You have to move the services to “the cloud”

THE “CLOUD”

- Machines outside your LAN
- In the past they probably hosted your website
- Options
  - Physical computers
  - Virtualized computers
  - System Images (Docker)
  - Only code (Serverless)

SECURITY

SECURITY

- Think about security from day 0
- Your service(s) will be accessed from outside the LAN
  - Meaning: Internet
- Never expose your database server
- Use REST libraries with **known** security implementations
  - Use always **JWT** as a token that contains client side information
  - Learn all about **JWT** and its use

MIGRATION

MIGRATION STRATEGY

- ➔ Assess your applications
  - ◆ Modern apps (microservices, cloud native apps, etc...)
  - ◆ Legacy apps
  - ◆ Data
- ➔ Assess your applications workload
- ➔ Estimate migration costs
  - ◆ Cloud provider costs (workload, etc...)
  - ◆ Data exchange costs
- ➔ Decide what to migrate: prioritize!

WHAT TO MIGRATE

- Data
  - Synchronization costs and time
- Authentication services
  - Security and performance concerns
- IoT gateways
- UI services

PRIORITIZE

1. Migrate the data
2. Services for Web applications
3. Services for Mobile applications
4. Services for Desktop applications (possibly)

CLOUD PROVIDERS

PROVIDERS

- Local providers
  - They are... local...
  - Usually more expensive (I've found)
  - Less features on their platforms
- Amazon AWS
  - They invented the cloud as a service as is it
  - Packed with tons of features
  - Outstanding bandwidth
  - Management it's a bit too complex

PROVIDERS

- Microsoft Azure
  - Fast growing service (but still 2nd)
  - Packed with tons of features
  - Management it's a bit too complex
  - I guess it's the first choice for Windows based services (?)
- Google Cloud
  - Last of the big names
  - Packed with features
  - Good interface

PROVIDERS

- My advice: reach for the best
  - In terms of features
  - In terms of price
- Today the two contenders are
  - Amazon AWS
  - Microsoft Azure
- My choice
  - Amazon AWS

INFRASTRUCTURE

OPERATING SYSTEMS

- Windows
  - First choice for a Delphi developer
  - A must if you plan to migrate old technology directly to the cloud
  - A bit more expensive (of course)
- Linux
  - Rules the “cloud world”
  - Every tech that comes out the cloud is: first Linux
  - Very easy to manage Linux VMs
  - Remember: Delphi can build Linux applications

DELPHI & LINUX

- ➔ Only 64 bit non UI application (x86 platform)
- ➔ All that you need to build multi-tier/microservices apps
  - ◆ RTL
  - ◆ FireDAC library
  - ◆ REST libraries
  - ◆ Indy libraries...
- ➔ In Delphi no supports to build Linux daemons
  - ◆ Check-out my GitHub project
  - ◆ https://github.com/paolo-rossi/linux-daemon

DOCKER

- ➔ Born to solve problems that Delphi (developers) don't have
- ➔ Delphi creates executables with no dependencies
  - ◆ Only the ones that you want
- ➔ Delphi apps deployment it's very straightforward
- ➔ Docker it's not a full featured virtual machine
- ➔ Ideally only one "thing" can run
  - ◆ Even essential OS's services counts
- ➔ It's not recommended to "dockerize" an entire Linux machine

SERVERLESS

- ➔ The “new” thing in the cloud world
- ➔ You don’t buy a “server” but resources consumed by your code function (piece of code running)
- ➔ Wide support for interpreted or hosted code
- ➔ Not so great support for compiled code (C, C++, Delphi, etc...)

THANK YOU

wintech italia