

ARCHITECT'S APPROACH

G. Molines
2018-2019



ARCHITECTURE VIEWS



Views – how do you describe a car?



- Stainless Steel Body Panels
- Counter Balanced Gull-Wing Doors
- 4-Wheel Independent Suspension
- Rack & Pinion Steering
- Electronic Ignition System
- Body Side Molding
- Tinted Windows
- Intermittent Windshield Wipers
- Epoxy Coated Corrosion Resistant Frame
- Dual Braking System
- Halogen Headlamps
- Engine/Luggage Compartment Lights
- Interior Hood & Engine Compartment Release

REAR MOUNTED 174 CID OHV V6 ENGINE
5-SPEED MANUAL TRANSMISSION
BOSCH K-JETRONIC FUEL INJECTION
POWER ASSISTED 4-WHEEL DISC BRAKES
AIR CONDITIONING
AM/FM STEREO RADIO w/CASSETTE
POWER WINDOWS
CENTRAL DOOR LOCKING SYSTEM
TIPT & TELESCOPIC STEERING COLUMN
DUAL ELECTRIC REMOTE SIDE VIEW MIRRORS
CAST LIGHT-ALLOY WHEELS
GOODYEAR RCT STEEL BELTED RADIAL TIRES
ELECTRIC REAR WINDOW DEFROGER
ELECTRIC TACHOMETER

U.S. PROCESSING CHARGES

\$800.00

OPTIONAL EQUIPMENT

NONE

METHOD OF TRANSPORTATION	
TRUCK	MODEL YEAR
	1981
	\$25000.00

THIS LABEL HAS BEEN AFFIXED TO THIS VEHICLE PURSUANT TO THE REQUIREMENTS OF 15 U.S.C., §1231 ET SEQ., WHICH PROHIBITS ITS REMOVAL OR ALTERATION PRIOR TO DELIVERY TO THE ULTIMATE PURCHASER.

*GASOLINE, LICENSE AND TITLE FEES, STATE AND LOCAL TAXES ARE NOT INCLUDED IN THE MANUFACTURER'S SUGGESTED RETAIL PRICE.

FUEL ECONOMY RATING	
MODEL: 1981 DE LOREAN, 174 CID ENGINE, 6 CYLINDERS, SPEEDSHIFT MANUAL TRANSMISSION, MECHANICAL FUEL INJECTION, CATALYST EQUIPPED, FEED BACK FUEL SYSTEM.	
ESTIMATED MPG FOR COMPARISON	
THE ESTIMATED MILEAGE FOR THIS MODEL, 19, IS TO BE USED TO COMPARE CARS OF THIS MODEL WITH OTHER CARS. YOUR OWN MILEAGE MAY BE POORER DEPENDING UPON OPTIONS, DRIVING CONDITIONS, YOUR DRIVING HABITS, AND YOUR CAR'S OPERATING CONDITION.	
THE ESTIMATED MPG NUMBERS FOR OTHER SIMILAR SIZED CARS RANGE FROM 11 TO 24 MPG (AS OF JUN '81). BY COMPARISON, THE ESTIMATED MPG OF THIS MODEL IS 19. USE THESE NUMBERS TO COMPARE DIFFERENT MODELS. CONSULT THE GAS MILEAGE GUIDE FOR FURTHER INFORMATION.	
ANNUAL FUEL COST, \$1225 BASED ON 19 MPG, 15,000 MILES PER YEAR, \$1.55 /GALLON ASK THE DEALER FOR THE FREE 1981 GAS MILEAGE GUIDE TO COMPARE THE ESTIMATED MPG OF OTHER CARS. IT WILL TELL YOU HOW TO USE THESE NUMBERS.	



Role

- The architect
 - Collects needs:
 - Explicit requirements (← Product Owner)
 - Implicit requirements
 - State of the art
 - Constraints
 - Defines guiding principles
 - Team > all the way to > entreprise
 - Drafts the architecture views
 - Can explain his choices
 - Derives dependencies
 - Jump starts the team
 - Inception
 - Mock-up, prototypes
 - Framework selection

Collecting constraints

- Explicit:
 - derived from functional needs
 - costs
 - enterprise guidelines / charter
 - Performance, SLAs
 - Time (deadline, time to market)
 - security

Collecting constraints

- Implicit:
 - interfaces with existing systems
 - technology choices
 - hardware, platforms
 - partnerships
 - Company culture

Deliverables

- The architecture views
 - Diagrams, but also text explaining the choices
- Prototype demonstrating at least one use case
 - Walking skeleton

Scenarios

- Suggest several potential scenarios
 - And evaluate them
 - Criteria (Eg: costs, feasibility, time)
- Challenge some of the constraints
 - Try to lift some
 - Prioritize the rest
- Derive risks

Migration

- When new system replaces old one, how to guarantee continuity of service?
 - Big bang migration
 - High risk, user acceptance
 - Progressive
 - Coexistence of two systems
 - Complexity
 - Cost
 - Can last months, even years

Architecture levels

- We'll choose these 3 levels for the rest of this class
 - System architecture
 - Application architecture
 - Technical architecture

SYSTEM ARCHITECTURE



System architecture

- Structures the system in applications that communicate together
- Describes the applications, the flows and the messages
- What is an “application”?
 - A defined executable entity offering services through an interface
 - Relatively standalone

In this phase

- Look at the system as a collection of black boxes
 - You need to know (or define), for each black box
 - Its technology
 - Its functionality
 - Its interfaces
 - Structure the system in terms of these boxes
 - But don't drill in each box (yet)

Focus on the flows

- Ensure completeness and consistency
- Adopt industry standards for:
 - Messages
 - Transport layers
 - Protocols
- For async messages, plan to use a MOM
 - (Message Oriented Middleware)

4-step approach

1/ Description of each application

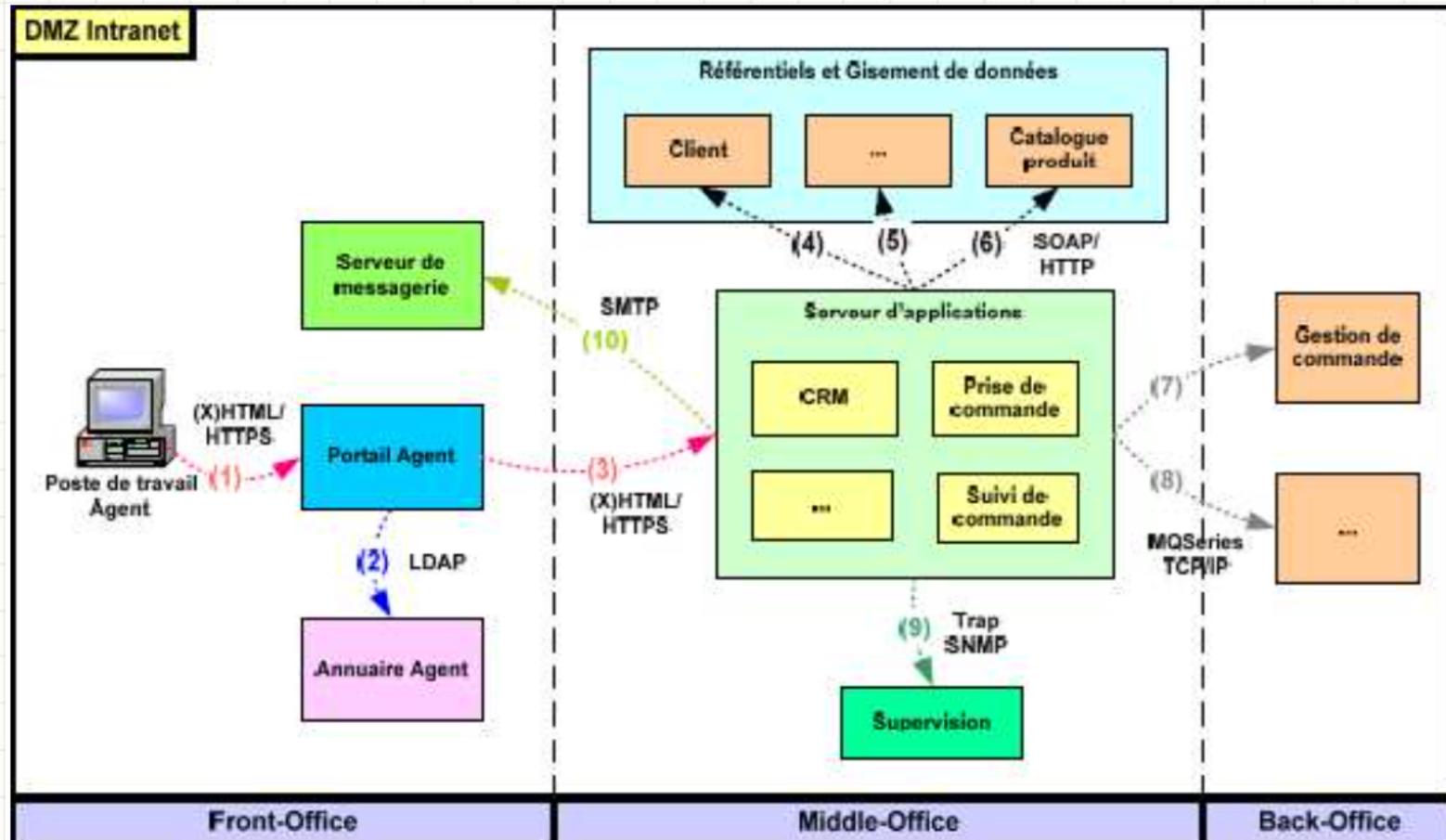
- Statically
- Functional
- Technical
- Include external systems you'll have to interact with
- Include infrastructure services
- Local technical constraints

4-step approach

2/ Application cartography

- A static view of the system
- Shows the flows and messages
- Don't forget external systems

Application cartography



4-step approach

3/ Message inventory

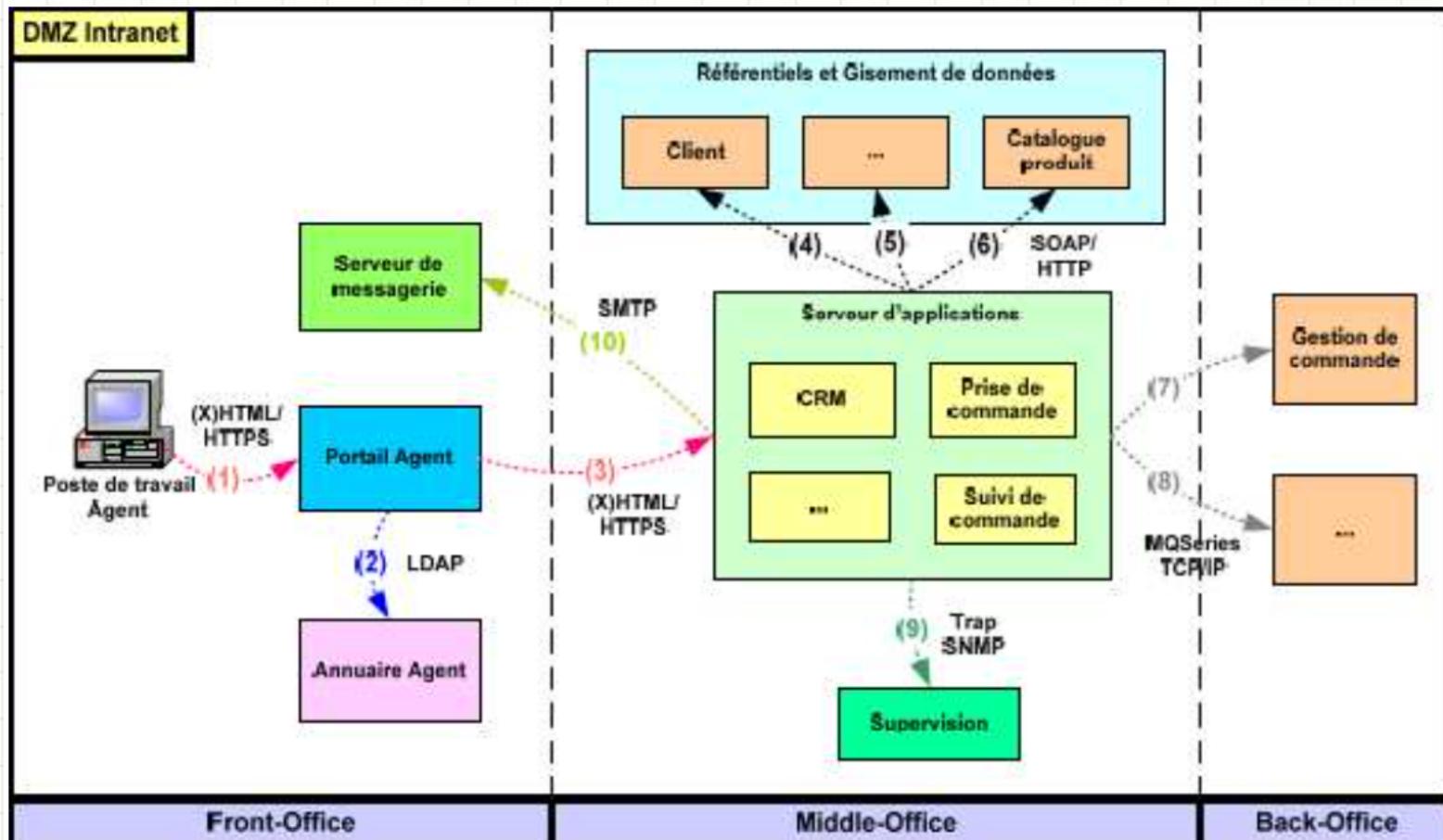
- Reference each flow, and for each of them
 - Direction
 - Physical layer
 - Transport
 - Message format
 - Protocol
 - Security

4-step approach

4/ Dynamic aspect

- For some (not all) use cases
- Show the order of flows
- Number their sequence on block diagram, or build sequence diagrams
- Check for consistency

Dynamic aspect



APPLICATION ARCHITECTURE



Application architecture

- Purpose:
 - Structure each application from its specifications
 - Layers, frameworks, design patterns, components, technical services
- Now we open each box – white box approach

Application architecture

- Language choice
- Libraries
- Internal structure and packaging
- Tiers

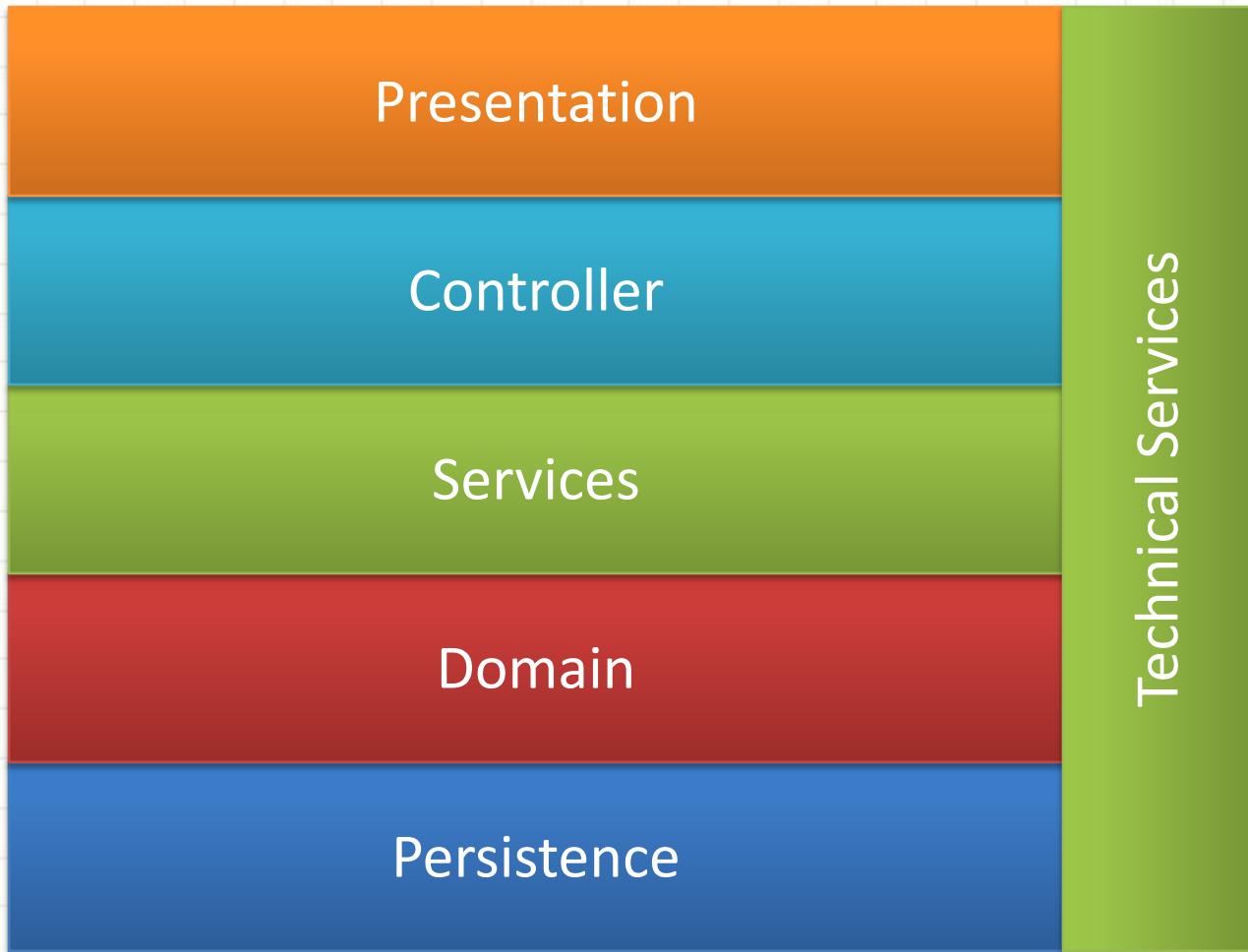
Approach

- The architect defines the application frameworks and structure
 - Documents it
 - Ensures the dev team understands and applies it
- As well as tools (dev env)
- And dev practices
 - See software engineering class on test, integration, etc.

Application structure

- Layer view
 - Logical view showing the grouping of the application function
- Tier view
 - Physical view, showing where each application function runs

Layer view - example

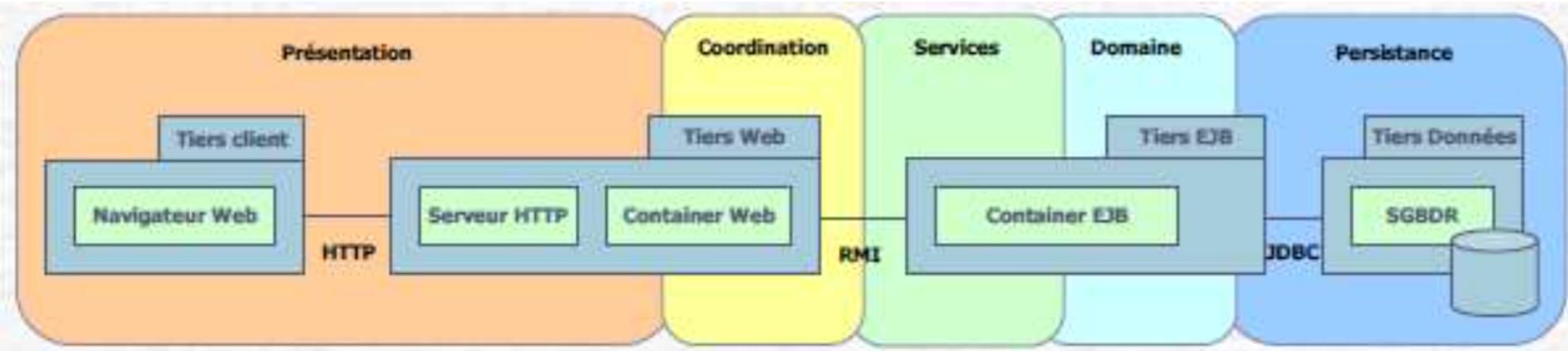


Tiers

- 1-Tier
 - All layers run as a single binary on one system
 - Example: Microsoft Word
- 2-Tier
 - Typically, a binary on one system, plus a database on another
 - DB reconcile users
 - Poor ability to scale and handle concurrency
 - But an important step in software history

Tiers

- n-Tiers
 - Now most common, Esp. for distributed applications, which are now standard



Architect approach

- Also take into account the system constraints
 - Security
 - Performance
 - High availability
 - Etc.

TECHNICAL ARCHITECTURE



Technical architecture

- Defines the infrastructure
 - Machine, servers
 - Communication infrastructure
 - Base software: OSes, DBs, MOMs, etc.

Approach

- Describe physical infrastructure
- Define QoS
- Sizing
 - Requires benchmarking
- Monitoring
- Cost

OPERATIONS



Environments related to the system

- Development
- Integration
- Benchmarks
- Testing
- Training?
- Pre-prod
- Production

Operations

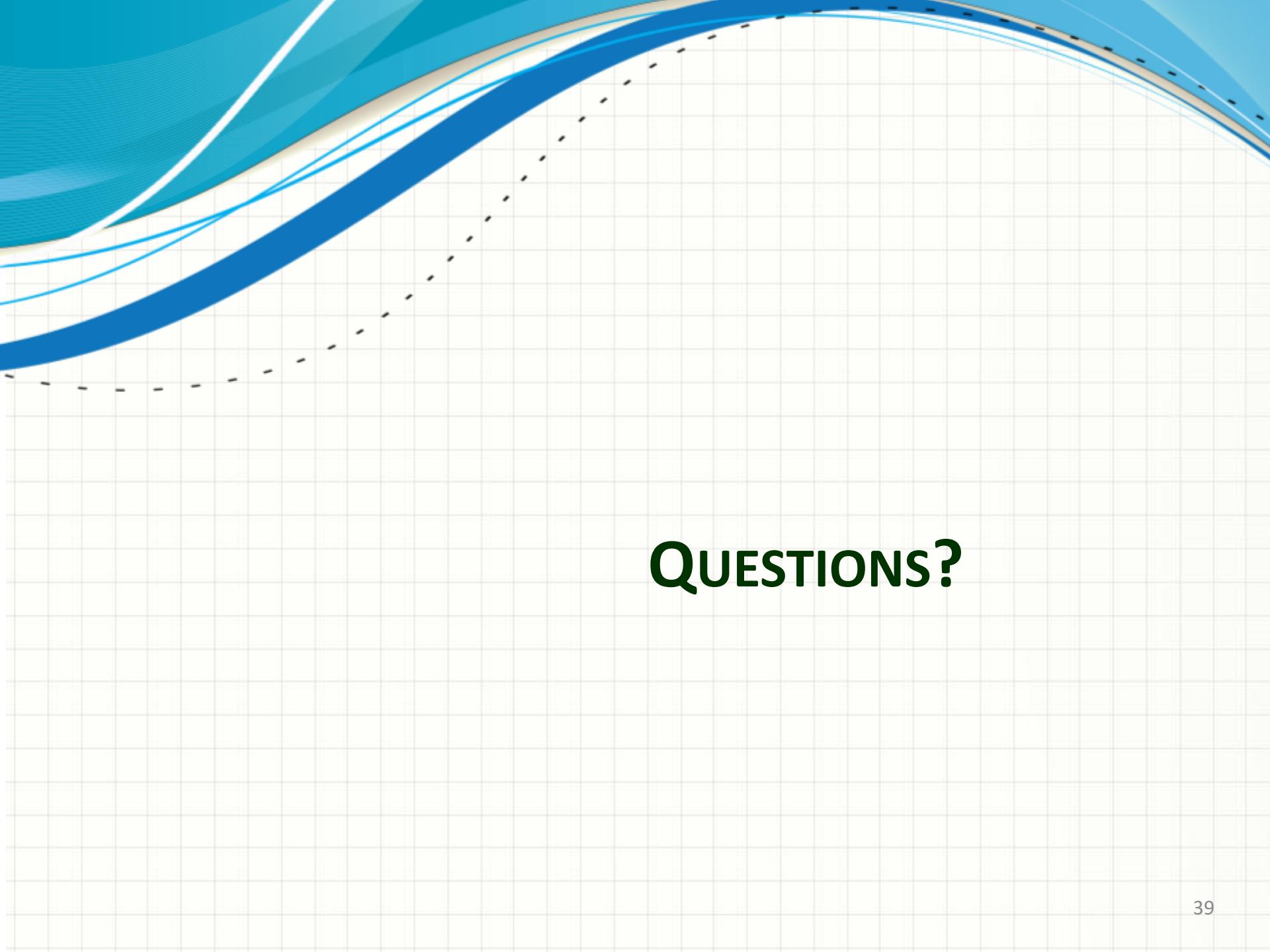
- How is it organized?
- Backups
 - Data
 - Power
 - Teams
- System supervision
- User support

ATTITUDE



Architect's attitude

- Humility
- Adaptability
- Curiosity
- Try to understand
 - Explore, get your hands dirty, evaluate
 - Then only decide
 - Not the other way around



QUESTIONS?

After the break....

- For our smart transportation project, what to do?
 - Scope – done, I'll give you feedback
 - One *component* diagram
 - What technology do we master / do we need?
 - Team roles
 - Rough plan → **deliverable for Sunday, Oct 7th.**
 - Create Epics
 - With success criteria
 - Break down Epics in User Stories
 - Development environment
 - Go !

APPENDIX