**DEMO**:

-1H PRERUN PWS 00 ON THE CLOUD AND TEST IT

-1H PRERUN PWS 02 06 EUREKA ON THE CLOUD AND TEST IT

-1H PRERUN LOCALLY

Scenario docker container - stop

scenario base transaction - stop

scenario eureka

13/09/2016 TODO HOME

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Office\14.0\Registration\{90140000-0011-0000-0000-0000000FF1CE}

Product ID: 82503-566-0254782-48558

Product Type: 566 (Unknown)

Serial Number: 7QTXM-MV9FC-7MVDJ-T7XWK-9PCMV

1. ~~Check project 00 first demo – please align logic to 00PWS~~
2. Check deployment 00PWS and run it OK
3. ~~Prepare a fake deployment project PWS su pivotal~~
4. Launch fake deployment
5. PWS:DEPLOYMENT EUREKA-SERVER-SERVICE-CONSUMER OK
6. ~~PWS:CHECK ERRORS ON DEPLOYMENT EUREKA- SERVER- SERVICE – COMUSUMER~~
   1. Ok web 06 consumenr
   2. ~~Fix starter version 02 06 service4eureka~~
   3. ~~https://github.com/spring-guides/gs-service-registration-and-discovery/blob/master/complete/eureka-service/src/main/resources/application.properties~~
   4. ~~Check new deploy on 06 service check booking class 000077.999~~
   5. Bind database pws on tool
7. CHECK DEMO TRANSACTIONAL BEHAVIOUR on pc ram consumption
8. CHECK DEMO EYUREKA SERVICE DISCO BEHAVIOUR on pc ram consumption
9. Detailed how eureka server works (notes on home’s table)
10. <http://stackoverflow.com/questions/33921557/understanding-spring-cloud-eureka-server-self-preservation-and-renew-threshold>
11. <https://www.google.com/maps/d/viewer?hl=it&authuser=0&mid=1KXbNWEmT194EMIVopTnmVTihcyA>

Complete:

1. Scenario local
2. Scenario docker
3. Scenario docker Hub
4. Scenario PWS
5. Scenario Transactional
6. Scenario Eureka LOCAL
7. Scenario Eureka PWS

Sommario

[1.1 [REQUISITI 1 – FUNZIONALI ] 7](#_Toc462308451)

[1.2 [REQUISITI 2– non FUNZIONALI ] 7](#_Toc462308452)

[1.3 [REQUISITI 3– project management ] 7](#_Toc462308453)

[1.4 [technology stacks] **Errore. Il segnalibro non è definito.**](#_Toc462308454)

[1.5 Lifecycle vs service vs environment specifity **Errore. Il segnalibro non è definito.**](#_Toc462308455)

[1.6 [MICROSERVICES database per service : DESIGN PATTERN] **Errore. Il segnalibro non è definito.**](#_Toc462308456)

[Microservice and Database per service will fulfill the non functional requirements 7](#_Toc462308457)

[1.7 [SYSTEM LANDSCAPE] 9](#_Toc462308458)

[2 Demo 10](#_Toc462308459)

[2.1 [DEVELOPMENT / UNIT TEST ] 10](#_Toc462308460)

[2.1.1 LOCAL H2 10](#_Toc462308461)

[2.1.2 LOCAl MYSQL 10](#_Toc462308462)

[2.2 [SCENARIO QUALITY ASSURANCE/INTEGRATION TEST –DOCKER LOCAL] 11](#_Toc462308463)

[2.3 [SCENARIO QUALITY ASSURANCE TEST –DOCKERHUB – JENKINS@OPENSHIFT] 12](#_Toc462308464)

[2.4 PWS on COlud foundry 12](#_Toc462308465)

[3 Microservice Transactional scenario with event driven architecture 13](#_Toc462308466)

[4 Wiring microservices - Service Discovery 23](#_Toc462308467)

[4.1.1 02-eureka-server 23](#_Toc462308468)

[4.1.2 06\_bookABatterySERVICE4EUREKA 24](#_Toc462308469)

[4.1.3 Discovery 26](#_Toc462308470)

[06\_bookABatteryCLIENT\_DISCOVERY\_SERVICE 26](#_Toc462308471)

[4.2 Load balancing with ribbon 27](#_Toc462308472)

[4.2.1 06\_bookABatteryCLIENT\_FEIGN\_SERVICE 27](#_Toc462308473)

[4.3 Load Balancing **Errore. Il segnalibro non è definito.**](#_Toc462308474)

[4.4 Load Balancing 30](#_Toc462308475)

[5 Backup 35](#_Toc462308476)

**[PRESENTATION OF THE WORK]**

As from the title the aim of this work is to show a possible end to end application lifecycle development process of a microservice ecosystem (architecture) form the collection of requirements to delivery passing by the steps of design with the choice of patterns and the corresponding technology, then the development, the integration and the quality assurance.

~~try to show a possible process of development of a microservices ecosystem from the very first phase of design to delivery.~~

~~I'm going to show a possible end to end application  lifecycle development process  of a microservice architecture from requirements to delivery passing by steps of design integration and qa (phases).~~

~~In will try to face a possible scenario of developmebt  
End to end process from design to delivery~~

~~Facing the choose of patterns and the corrisponding technology yhat fulfull the requirements in respect of requirements  
And at the same time the right lifecycle in respect the project  management requirements~~

**[ABSTRACT - BUSINESS CONTEXT]**

~~Let us see the business context thata originated the need of application development~~

~~The possible business context for such a computing system.~~

So let us see in brief the business context that originated the need of this application development

[FIRST IMAGE OF THE MAP OF ROME – HISTORICAL DISTRICT]

This is historical district of Rome the city where I live since I was born.

It is a very ancient area whose urban development goes back in the centuries starting from the roman age, more than  2000 years ago, going over the middle age and the reinassance.

[SECOND IMAGE – HISTORICAL DISTRICT OF ROME]

The blue line surround the city motorvehicle limited traffic zone applied form authority since many years ago. It is an area of about 10 square kilometers caracterized by a strong concentration of historical monuments, tourism facilities and national institutions (government and parliament), that generate a strong demand of mobility.

As a matter of a fact the urban development was not planned for the actual need of people and motor vehicle mobility. The street are often narrow in a context of a fragile architectural environment.

[THIRD IMAGE – SMART ELECTRIC VEHICLE]

A possible solution to the mobility challenge for the reduced spaces could be carry out introducing a smart electric vehicle for goods delivery, able to realize an efficient delivery service in respect of architectural environment but also of pollution of the air.

[FOURTH IMAGE – SMART ELECTRIC VEHICLE – BATTERY ]

To overcome the limited life of the batteries these vehicles has been engineered with a special battery pack that could be easily and rapidly changed in dedicated service station.

In this way this vehicles will not need to stop to charge theirs batteries and so could be much more efficient in comparison with the other electric vehicles.

[FIFTH IMAGE – STATION ]

I do not know if this is a realizable scenario but if so it is necessary a software solution by means manage it.

The requirements are that Every driver according to his delivery plan program each morning the expected battery changes.  
This by means of a mobile application he will book one or  more pit stops for a fresh batteries supplied by the stations.

Besides in case of emergency it will be the vehicle itself that will notify to the driver the necessity to come to the nearest station with available batteries.

So these vehicle could lead an ideal  24/7 service without charging stops in comparison with those vehicles not engineered in such a way.

## [REQUIREMENTS 1 – FUNCTIONAL ]

## [REQUIREMENTS 2– NON FUNCTIONAL ]

## [REQUIREMENTS 3– PROJECT MANAGEMENT ]

## [REQUIREMENTS FULFILLMENT – MICROSERVICES DATABASE PER SERVICE PATTERN ]

## Microservice and Database per service will fulfill the non functional requirements

So a Microservices architecture will be the design pattern that accomplish the listed requirements (face the choose of a database design pattern)

What is the database design pattern that best fit a microservice architecture?

Description of database per service pattern for developing Micorservices

Database per service will fulfill the non functional requirements

## [REQUIREMENTS FULFILLMENT – TECHNOLOGY STACKS]

A possible solution of these requirements could be led by the following technology stack coming by the stìpring project abd Netflix

A possivle resolution of these requirement could be led by The follpwimg  texhnology stack spring and netflix based .

In this chart i have tried to summarize the technology stacks that I have used to realize the design patterns defined for choose for the development.

In this chart I have tried to summarize the technology stacks that realize the design patterns choosed for the architecture of this system

SPRING BOOT RUNNABLE FROM COMMANDLINE EXAMPLE

In this chart i have posizionato the    that accomplish the lifecycle steps imagined for this project and the corrispondending realizing services  
both in a local and in a cloud environment

## [REQUIREMENTS FULFILLMENT – LIFECYCLE ]

In this chart I have arranged that will accomplish the lifecycle steps imagined for this project and the corresponding realizing services both in a local and in a cloud environment.

In this chart i have arraged the    that accomplish the lifecycle steps imagined for this project and the corrispondending realizing services   
both in a local and in a cloud environment.

Lifecycle vs service vs env  
In this chart i have tried to summarize the technology stacks that realize the design patterns chosed for the development.

## [SYSTEM LANDSCAPE]

Face the design of one of the microswrvices of the system. That which records the booking information.

Face the design of one of the microservice of the system that which record the booking information

* Show picture (db/logic/api)

This is the simplest landscape of a microservices architecture. There is a simgle service who expose his business logic only by means of restfull web service api. During this speech i will add new seevices that realize other aspects of the system.

The landscape of this system will became muxh more real during this presentation adding new services or microservices talking about event driven architecture and wiring microservices

# Demo

**DEVELOPMENT::PRELOAD CHROME PAGES**



**INTEGRATION::PRELOAD**

1. **COMMAND LINE DOCKERMACHINE RUN**
2. **COMMAND LINE ready to run list of container**
3. **COMMAND LINE READY TO RUN CONTAINERS** 
   1. **MYSQL**
   2. **APP WITH LINK**
4. **CHROME PAGES**
   1. [**http://192.168.99.100:7111/bookABattery/list**](http://192.168.99.100:7111/bookABattery/list)
   2. [**http://192.168.99.100:7111/bookABattery/addBooking/STAZ001/BATT001/ROMA/42/16**](http://192.168.99.100:7111/bookABattery/addBooking/STAZ001/BATT001/ROMA/42/16)

**QUALITY ASSURANCE::PRELOAD CHROME PAGES**

**PRODUCTION::PRELOAD CHROME PAGES**

## [DEVELOPMENT / UNIT TEST ]

**INTRODUCTION TO THIS PHASE**

Besides the implementation details of the frameworks in this first phase of lifecycle, development,

We see the resolution if the database backing service

Keeping the same coding both for a local in memory He , MySql

* Spring boot and services binding
  + Automatic behaviuor in cloud foundry
  + The goal should be a smooh transition from local execution form developer desktop to production in Cloud Foundry
  + Binding realized with declarative approach
    - Application.properties
    - Yaml file
    - [show the differences]
    - Dual Running: Local with H2/mysql
    - And the in the Cloud with MySQL
    - Configuration
  + Different file
* H2 in memory database
* <https://spring.io/guides/gs/accessing-data-rest/>
* <https://spring.io/guides/gs/accessing-data-jpa/>
* <http://blog.netgloo.com/2014/10/27/using-mysql-in-spring-boot-via-spring-data-jpa-and-hibernate/>

**[SHOW PPT SLIDES WITH IMPLEMENTATION DETAILS]**

### LOCAL H2

00pwd PROFILE h2

Environment information provided by spring boot

<http://localhost:7111/h2-console/>

<http://localhost:7111/trace>

<http://localhost:7111/flyway>

<http://localhost:7111/metrics>

example /flyway number of request

<http://localhost:7111/bookABattery/list>

list of booked batteries – EMPTY

<http://localhost:7111/bookABattery/addBooking/STAZ001/BATT011/ROMA/42.984545454/16.74444/>

add one or more booking

<http://localhost:7111/bookABattery/list>

list of booked batteries – LIST EVIDENCE

<http://localhost:7111/findNearestStation/41.90231/12.4832/0.50/>

find nearest station for an emergency pit stop

### LOCAl MYSQL

00pwd PROFILE mYSQL

[**http://localhost:7111/health**](http://localhost:7111/health)

**show the Jason with information about the disk the up status and the database in this cas MySQL**

<http://localhost:7111/bookABattery/addBooking/STAZ001/BATT011/ROMA/42.984545454/16.74444/>

add one or more booking

<http://localhost:7111/bookABattery/list>

list of booked batteries – LIST EVIDENCE

<http://localhost:7111/findNearestStation/41.90231/12.4832/0.50/>

find nearest station for an emergency pit stop

### [SCENARIO ~~QUALITY ASSURANCE~~/INTEGRATION TEST –DOCKER LOCAL]

**00DLoc\_bookabattery\_local\_docker.docx**

**So let’s go ahead in the develipong process facing the phase of integration test**

**The commit master will have the duty of conduct the integration test against all development done on one or more sevices modified for the new system release or for bug fixing coming from the QA or production environments.**

**THE INTEGRATION TEST WILL BE CONDUCTED IN AN EVIRONMENT WHERE THERE WILL BE AVAILABLE ALL THE SERVICES needed for the system to be consistent.**

**In a simple scenario two or more developers each responsible for the development of a part of a microservice finish with success the unit test**

**Beside the skill of the qa team must be enough to get the application and run it, it is stated in the requirements that in this team there will be no need of technical slìkill that will manage the deployment or other technical tasks.**

**So the commit master in a dedicated branch deliver the work of the developers merging all the contributes and build the microservice**

Docker definition

-I have provided an empty database image that will be load by flyway at start up

Show Dockerfile

Show dockerfile template

Show maven implementation and plug in

Create a local docker image

Run the container + database

Show bindings

### [EVENT DRIVE DEMO]

Event driven  
Example if extra coding and infrastructure management.   
Show the effective overhead of coding and infra management in a simple example.  
Drawback  
Benefits

## [SCENARIO QUALITY ASSURANCE TEST –DOCKERHUB – JENKINS@OPENSHIFT]

**00D\_ScenarioQulityASS\_dockHub-GITHUB.docx**

Among the requirements it has been stated that the project’s teams

The progect management requirements shoud accomplish teams geographically distributed.

In such a context the delivery of the application as a docker machine could be realized by means of Docker Hub as a project’s images registry.

## PWS on COlud foundry

Database as service pattern will enforce database per service pattern.  Dedicated instance , dedicated schema and horizontal scale up

Show definition of application .cloudfoundry and its link to application.yml

Deploy by cf tool – maven (only show plugin)

Pws  
Database as service pattern will enforce database per service pattern.  Dedicated instance , dedicated schema and horizontal scale up

Database per service  pattern implementation inside a cloud env is natively implemented  
Segregation is enforced

The databasr is dedicated to the servuce and could be scaled out independently

YML

Db resolution show (?)

# Microservice Transactional scenario with event driven architecture

1. DESCRIPTION
2. Start ZOOKEEPER
3. Start KAFKA
4. Start MongoDB
5. START LISTENER TO TOPIC
   1. confirmBookingTopic
   2. notConfirmBookingTopic
   3. pendingBookingTopic
6. project **01-batteryBookingInformation\_SERVICE**
   1. DETAILS: TOPIC IMPLEMENTATION
   2. DETAILS: PROCESS DB INQUIRY
   3. DETAILS: POM
      1. spring-cloud-stream
      2. spring-cloud-starter-stream-kafka
      3. spring-boot-starter-data-mongodb
   4. RUN FROM .BAT
   5. LIST BOOKING INFOS

<http://localhost:7112//bookingInfoMaterializedView/list> NULL

1. project **01-batteryManagement\_SERVICE**
   1. DETAILS: TOPIC IMPLEMENTATION
   2. DETAILS:
   3. RUN 01\_**batteryManagement** \_SERVICE
      1. Show log output
      2. Show database load
   4. Show topic subscriber **EMPTY**
   5. RUN FROM .BAT
2. project 01\_bookAbattery\_SERVICE
   1. DETAILS: TOPIC IMPLEMENTATION
   2. DETAILS: PROCESS DB INQUIRY
   3. DETAILS: POM
      1. spring-cloud-stream
      2. spring-cloud-starter-stream-kafka
   4. RUN 01\_bookAbattery\_SERVICE –profile localmysql FROM .BAT
      1. Show log output
      2. Show database load



* + 1. Show log file with executing jobs

DbPollingPendingCleaner Job -> 09/06/2016 16:29:13

DbPollingPending Job -> 09/06/2016 16:29:13

DbPollingPending Job -> 09/06/2016 16:29:23

DbPollingPending Job -> 09/06/2016 16:29:33

DbPollingPendingCleaner Job -> 09/06/2016 16:29:38

DbPollingPending Job -> 09/06/2016 16:29:43

DbPollingPending Job -> 09/06/2016 16:29:53

DbPollingPending Job -> 09/06/2016 16:30:03

DbPollingPendingCleaner Job -> 09/06/2016 16:30:03

DbPollingPending Job -> 09/06/2016 16:30:13

DbPollingPending Job -> 09/06/2016 16:30:23

DbPollingPendingCleaner Job -> 09/06/2016 16:30:28

* + 1. Show Topic subscriber



* + 1. Show database update



* + 1. Blah













1. Stop core services and check information services still running

# Wiring microservices - Service Discovery

In sun an architecture microservices can cooperate for a computing system ecosystem shoud wire each other

Wiring microservices.  
Loosing coupled requirements are not onky those involving tech aspects. In a ecosystem if microservice it is likely thata microservices need each othe wiring.  
Microservices do nita act by themself.  
Si the loosing coupled requirent is completely satisfied implementibg a service discovery a registry where are resolved........

Wiring by resftfull call baut to achieve decouplig it is necessary that service shoud be discocered.

Eureka server is on its turna microservice (run as a spring boot application)

1. START 02 EUREKA SERVER
2. SHOW CONSOLE EUREKA
3. START 06 7111
4. SHOW REGISTRATION
5. START 06 SERVICE CONSUMER
6. SHOW /LIST SERVICES
7. SHOW /

### 02-eureka-server

Overview of the project

File yml etc

Configurations of Eureka

Run project from command line .bat

Show console



#### Detail the message “THE SELF PRESERVATION MODE IS TURNED OFF.THIS MAY NOT PROTECT INSTANCE EXPIRY IN CASE OF NETWORK/OTHER PROBLEMS”

2016-09-08 14:52:35.431 WARN 8232 --- [a-EvictionTimer] c.n.eureka.PeerAwareInstanceRegistry : The self preservation mode is disabled!. Hence allowing the

instances to expire.

<https://github.com/ExampleDriven/spring-cloud-eureka-example/blob/master/eureka-server/src/main/resources/application.yml>

http://stackoverflow.com/questions/33921557/understanding-spring-cloud-eureka-server-self-preservation-and-renew-threshold

### 06\_bookABatterySERVICE4EUREKA

Overview of the project

Eureka directives on application .class

File yml etc

Run project from command line .bat

Check registration



#### 06\_bookABatterySERVICE4EUREKA 7113-7115

Overview of the project

Eureka directives on application .class

File yml etc

Run project from command line .bat

Check registration



### Discovery

### 06\_bookABatteryCLIENT\_DISCOVERY\_SERVICE

Code description

Run from .bat

Call:

## Load balancing with ribbon

### 06\_bookABatteryCLIENT\_FEIGN\_SERVICE

Show implementation

<http://sdpsvrsa094:7112/>

show requested on the tree instance



Stop two and how requested on the one running





Suppress warn and info on services display

## PWS Load Balancing

**[PWS DISCOVERY AND BALANCING]**

Predeployed application only one service

Pivotal ws disco and balance

requirements satisfaction about

simple Scale up/down

No downtime during scale up

No load balancing configuration

Resolution of instance made by pws mechanisms

Describe eureka console

1. Service
2. Consumer

Launch test of balancing showing that only one instance is replying

LIVE: Scale instance of service then wait enough time (**take time**)

Launch test of available instances until discovered

Launch test of balancing

http://06-bookabatteryclient\_discovery\_service.cfapps.io/listDiscovery

Launch test of balancing

**[EUREKA ALSO DEFINITION]**

server:

port: 8761

eureka:

numberRegistrySyncRetries: 1

instance:

hostname: localhost

client:

registerWithEureka: false

fetchRegistry: false

serviceUrl:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

server:

enable-self-preservation: true

Explaining the most important lines:

server:

port: 8761

Here, we are configuring the Eureka Server to run on port 8761. This is the default port and you can change, but you need to give this port to the clients later on.

eureka:

numberRegistrySyncRetries: 1

If you are running locally, there is a 2 to 3 minutes wait until fulling boot up. This happens because Eureka will be looking for peers. To disable this, set to 0 (although you should never do this in production)

client:

registerWithEureka: false

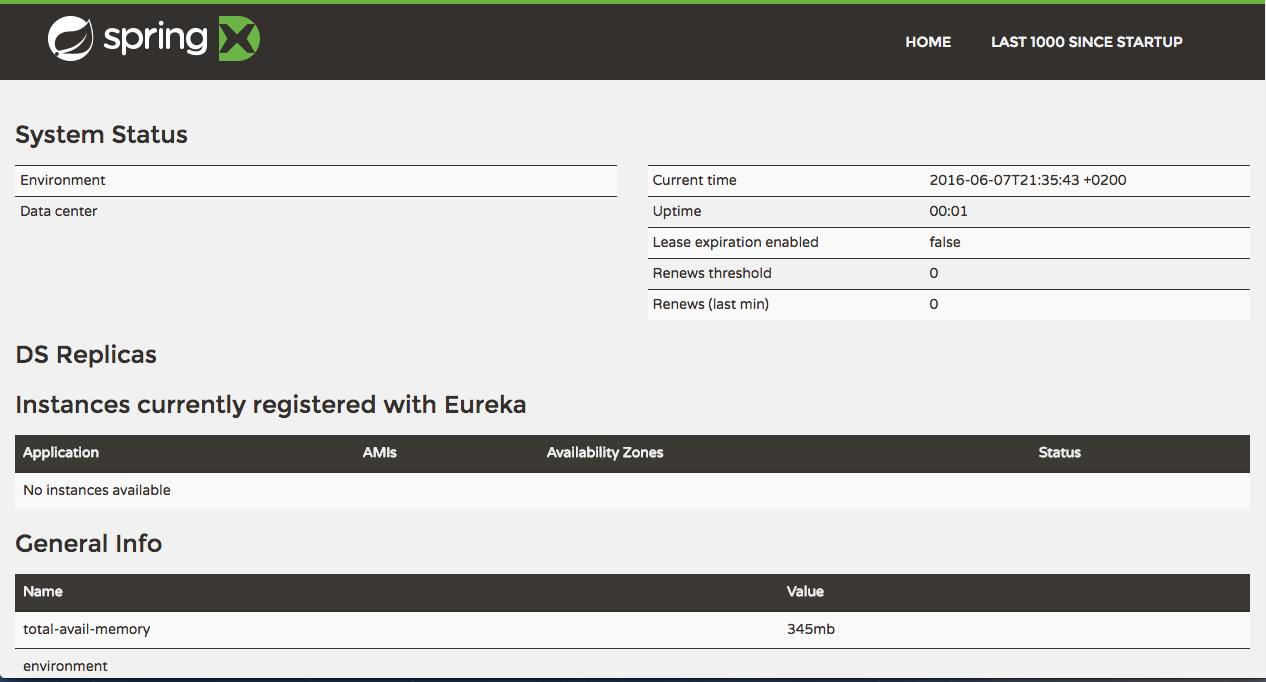
As this is the Eureka Server, we do not want it to register itself. Will always be set to false on server and true on the clients.

serviceUrl:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

the defaultZone is the fallback URL for every client that doesn't specify a preference for a server.

Now, the only thing we should do is enable the Eureka Server.   
To do so, go to the main class (at this point, you should only have one class in the project, though) and annotate it with @EnableEurekaServer, as follow



# Backup

The historic district of Rome  
(Prima sfide)  
Storia  
Romans  
Middle age  
900

So it is

A possible solution that realize both mobility and respect of environment could be led by smart electrical vehicle. Small that they can easily move among the small Street without no emission

The hystorical center of rrome is full of business activity, besides the seat of govermant institution so goods delivery is a critical   
To be much more

These vehicle colud lead an ideal  24/7 service without charging stops

Mobility sould be conducted in respect of ancient monuments and fragile buildings

Pit stop

Rome is the city where I live since I wos born. It counts about 3.5 official inhabitants to witch we have to add about 1 million of non resident people (students, tourists, workers). To these figures we have also to add about 700 vehicles each 1000 people, a rate that makes mobility a challenge.

Everyone could easily realize that with there figures besides mobility there are also pollution issues.

~~A partial solution of these issues could be addressed reducing the vehicles that daily deliver goods.~~

For a more sustainable for the environment AND RESPECTING THIS ARCHITECTONICAL VALUES and efficient mobility in such a context the daily delivery of goods could be done by means of (through) “smart” electrical vehicles.

~~To overcome the limited life of the batteries these vehicles has been enginnered with special battery pack that could be easily and rapidly changed in dedicated service station.~~

~~In this way these vehicle will not need to stop for charging their batteries and so could guarantee much more short breaks~~

~~Could be much more efficient in comparison to the other electric vehicle (that do non support this kind of batteries – not engineered in such a way)~~

I do not know if this is a realizable scenario but if so it is necessary a software solution by witch manage this scenario.

Each driver of this ecological delivery company according to his delivery plan each morning program the expected battery changes that will reasonably occur during a day. By means of a mobile application (web application) he will book one or more fresh batteries supplied by the stations distributed in the city.

In case of emergency it will be the vehicle itself that will notify to the driver the necessity to come to the nearest station with available fresh batteries, find by a software running in the vehicle.