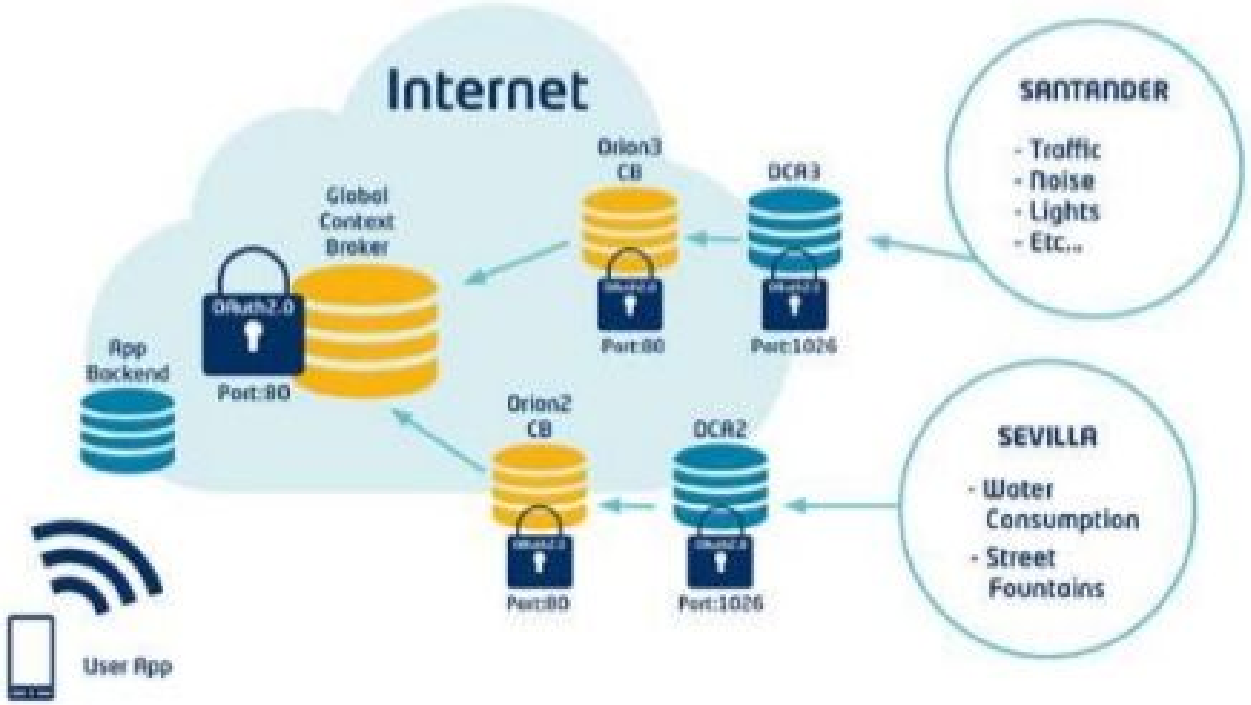
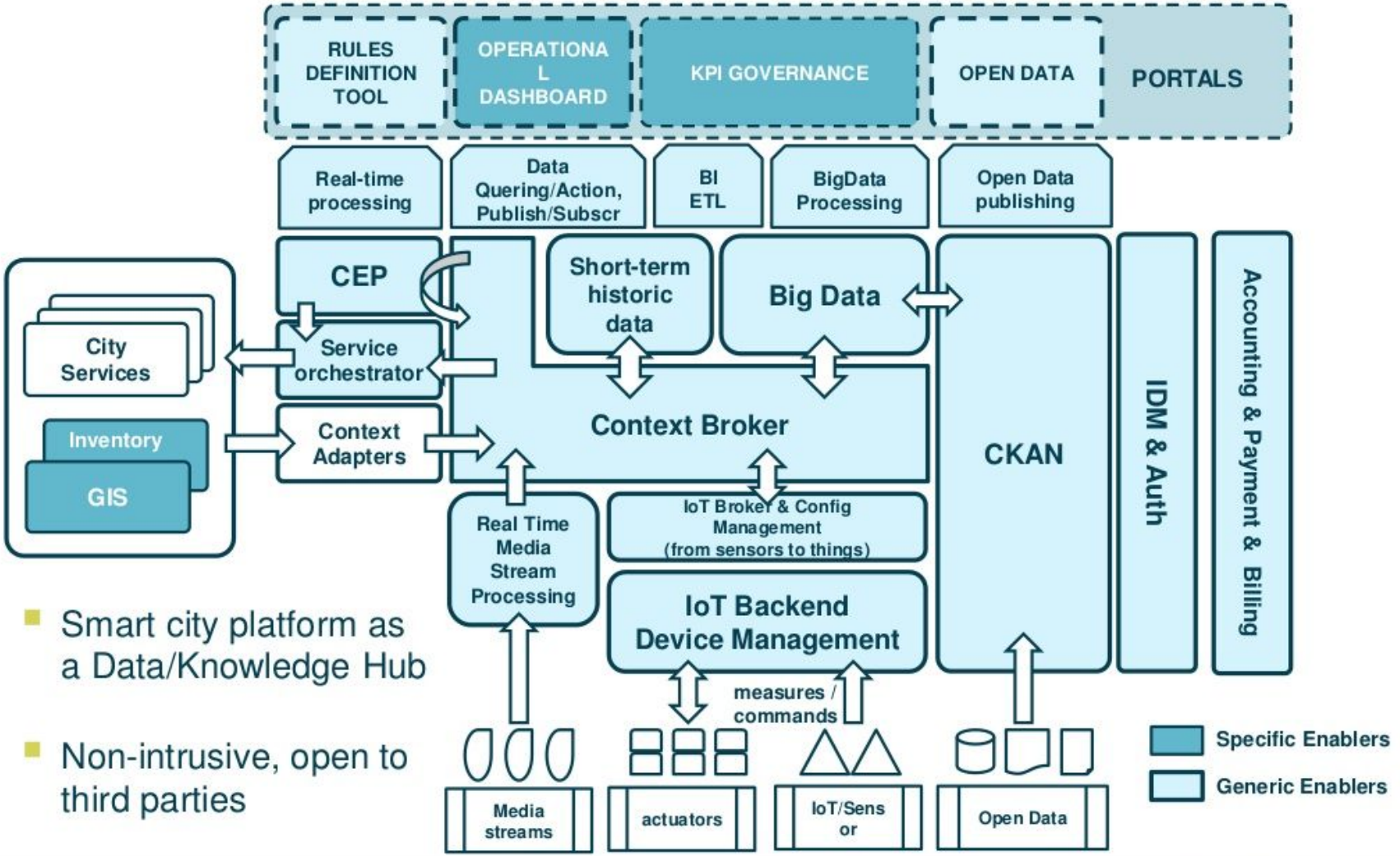
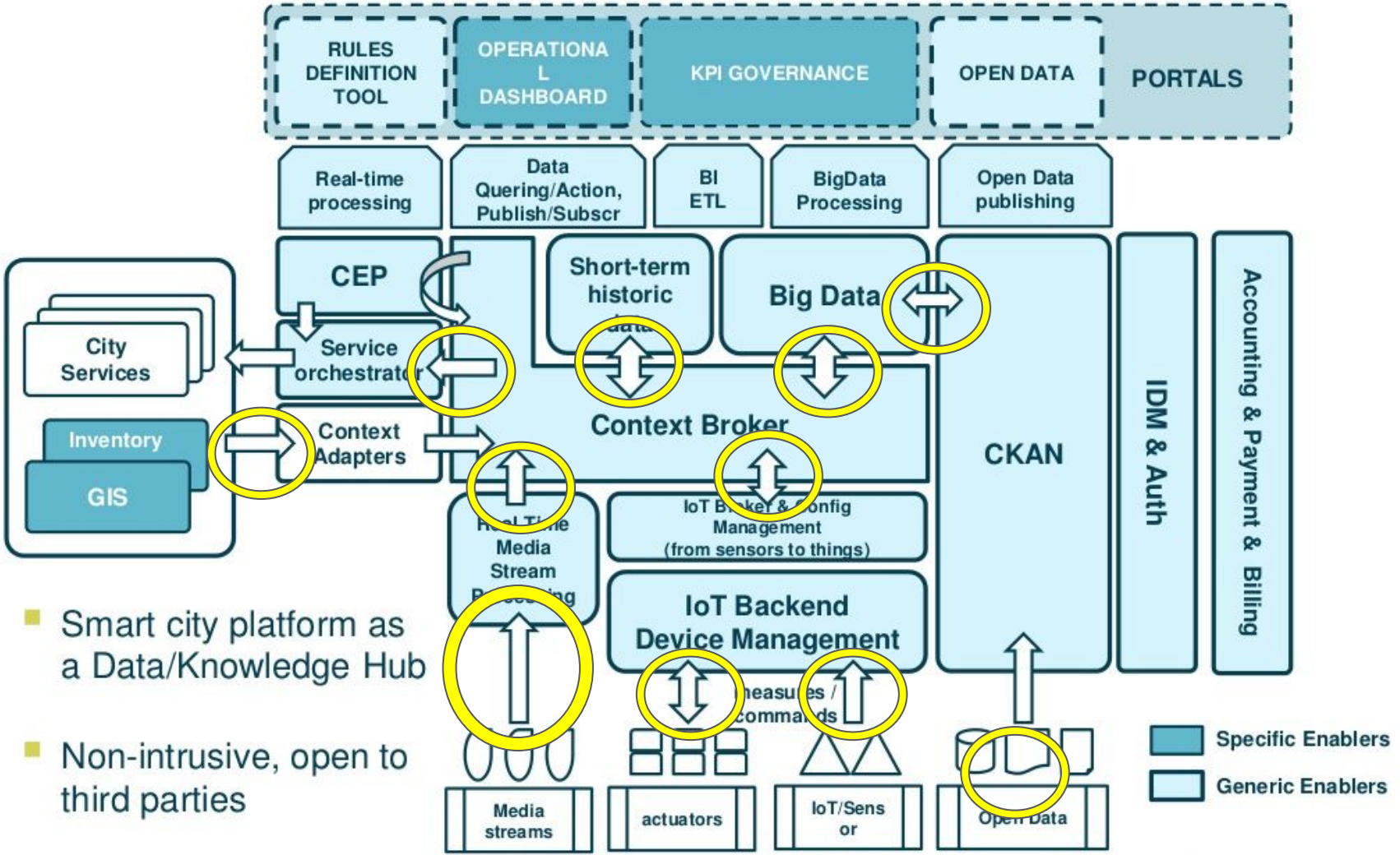
Kiara Advanced Middleware

KIARA Advanced Middleware is a Java based communication middleware for modern, efficient and secure applications.

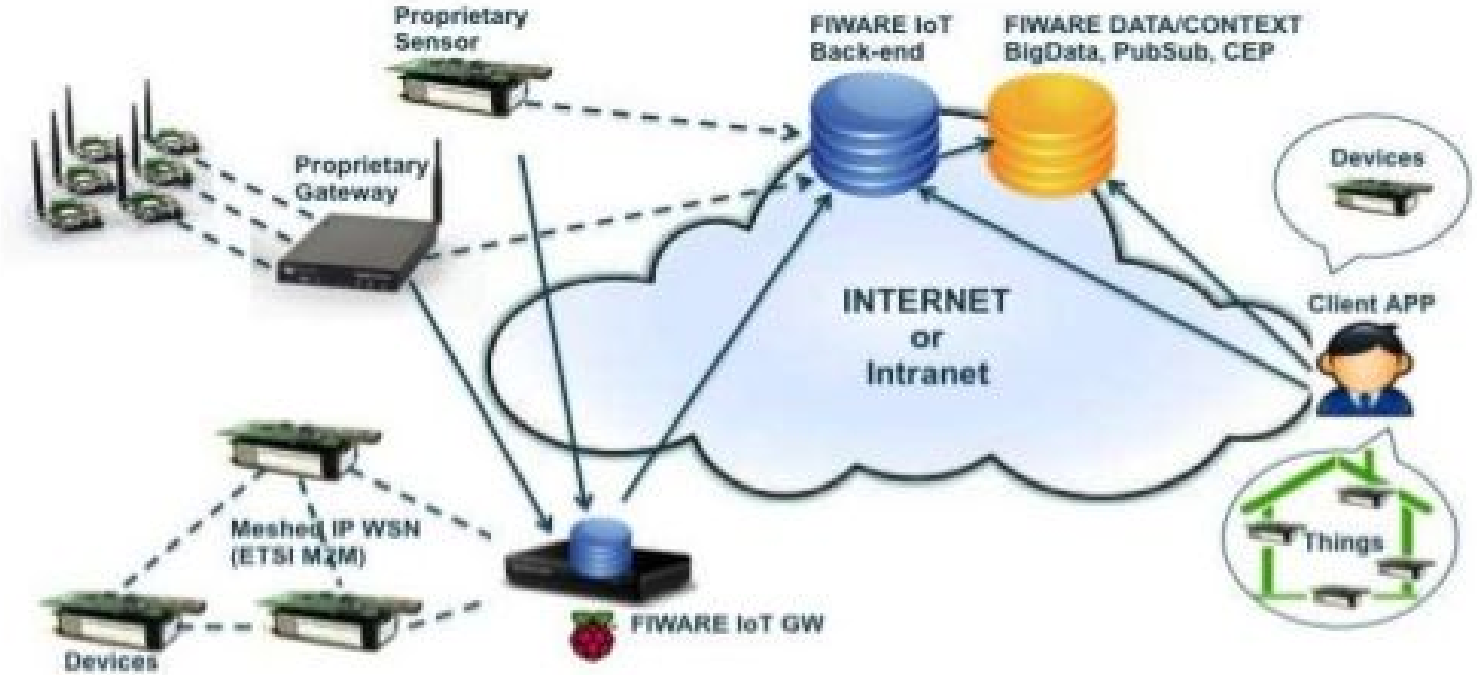




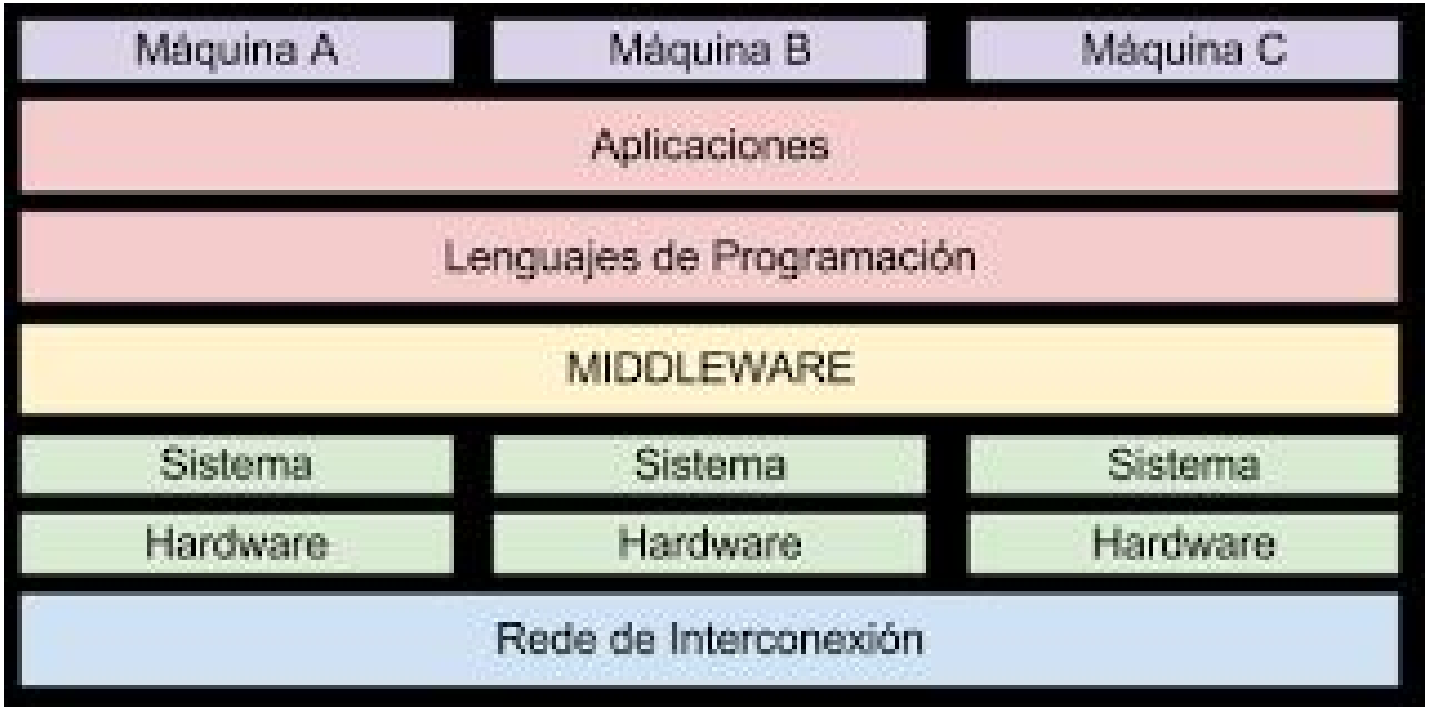
- Smart city platform as a Data/Knowledge Hub
- Non-intrusive, open to third parties



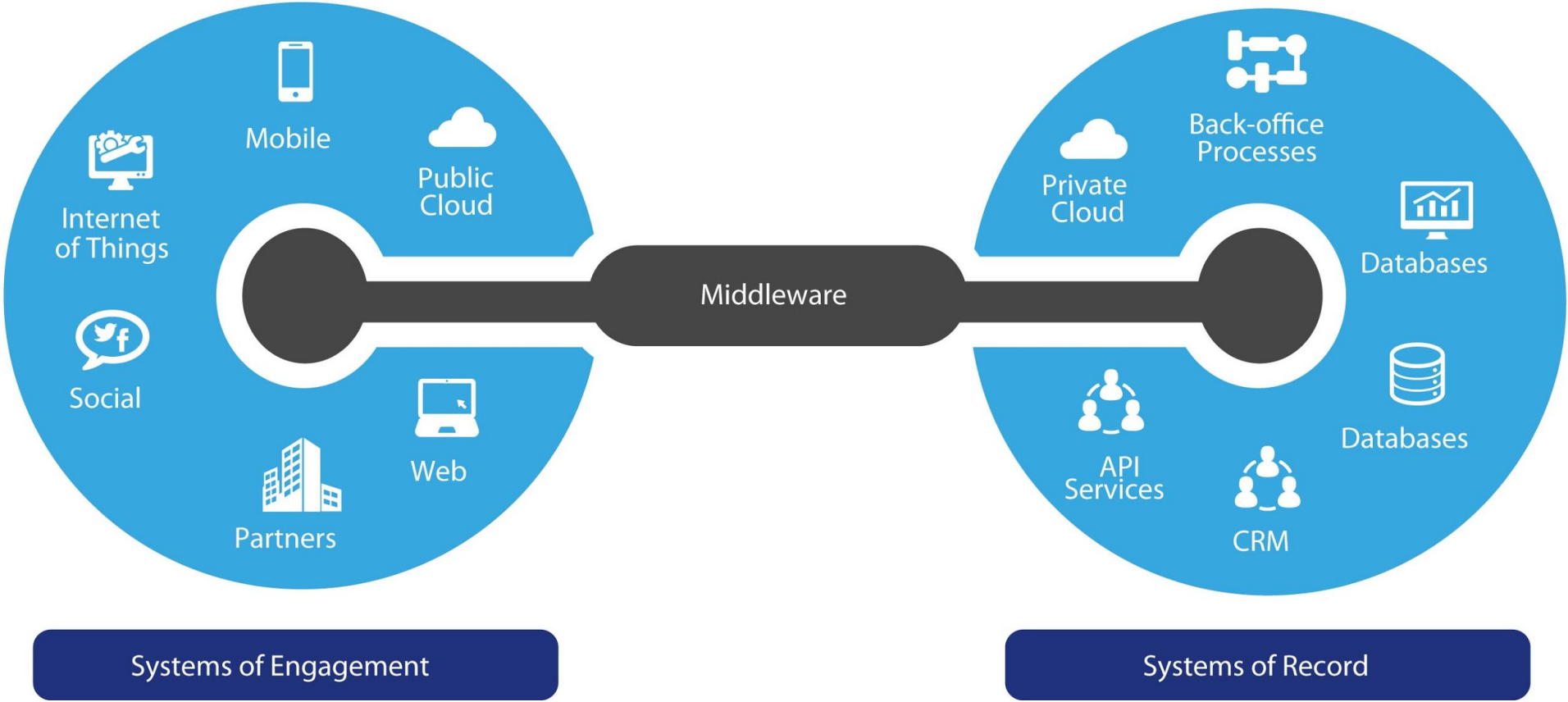
- Smart city platform as a Data/Knowledge Hub
- Non-intrusive, open to third parties


Proprietary
Sensor
FIWARE IoT
Back-end
FIWARE DATA/CONTEXT
BigData, PubSub, CEP
Proprietary
Gateway
Devices
INTERNET
or
Intranet
Client APP
Meshed IP WSN
(ETSI M2M)
Things
FIWARE IoT GW
Devices

=>facilitar o desenvolvimento de aplicações >> tipicamente as distribuídas => integração de sistemas legados <OU> desenvolvidos de forma não integrada automática.



PROCESSOS ou objetos em um grupo de COMPUTADORES, que INTERAGEM de forma a implementar COMUNICAÇÃO e oferecer suporte para COMPARTILHAMENTO de RECURSOS e aplicativos distribuídos.


Mobile
Public Cloud
Internet of Things
Social
Web
Partners
Middleware
Private Cloud
Back-office Processes
Databases
Databases
API Services
CRM
Systems of Engagement
Systems of Record

Easy to USE API

Permite Cahadas **Síncronas** e **Assíncronas**

Modern Interface Definition Language (**IDL**)

SUPPORT FOR THE **PUBLISH / SUBSCRIBE**

Modelo de Comunicação

Nas últimas Versões:

Advanced Security Features

Cripitografia e Autenticação

Modo **APPLICATION DERIVED AND MAPPED** => Declaração dinâmica de Funções e Tipos de Variáveis

Não precisa de serviço Rodando em Background

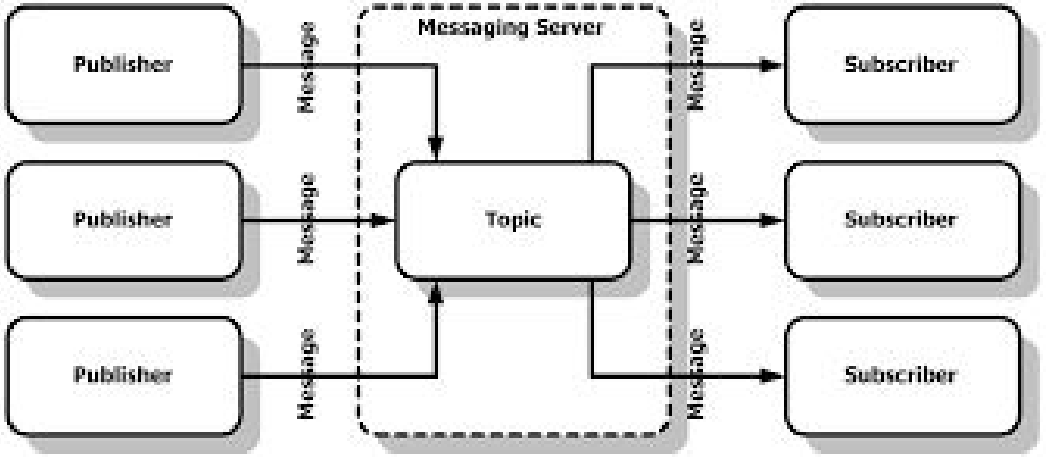
Implementa

- Transparent
- Fast
- Secure
- Standard Conform

COMMUNICATION

between

Aplicações ou Serviços



With

Friendly

AND

Extendable

API

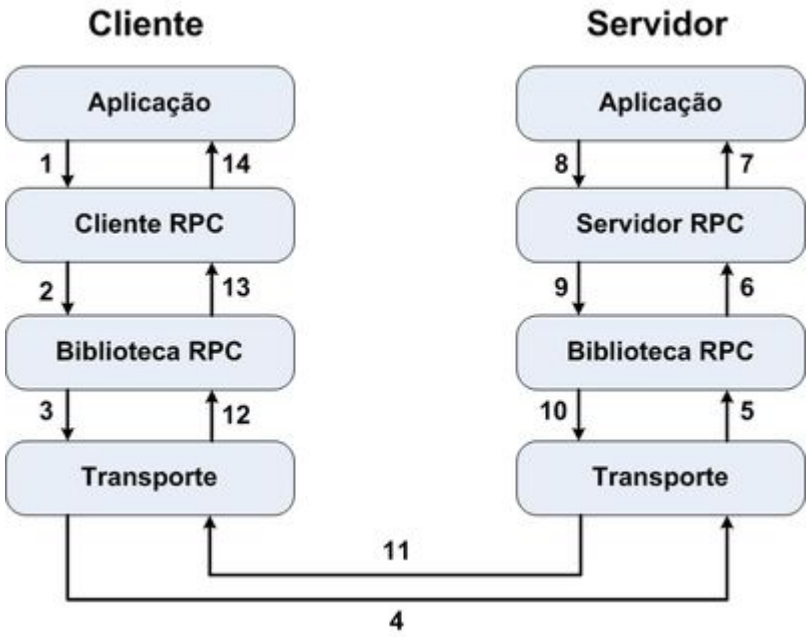
KIARA É UM RPC

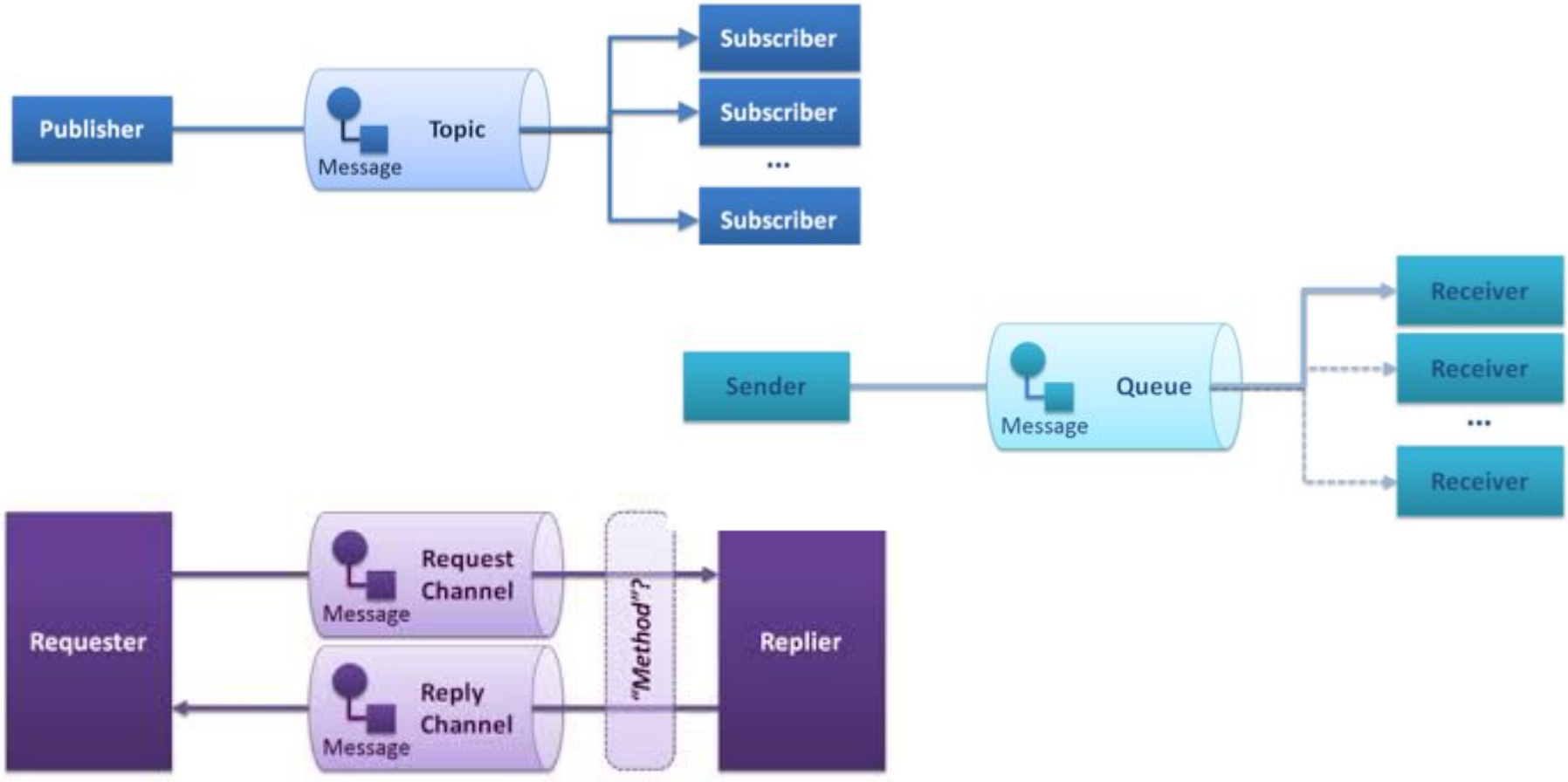
Remote procedure call

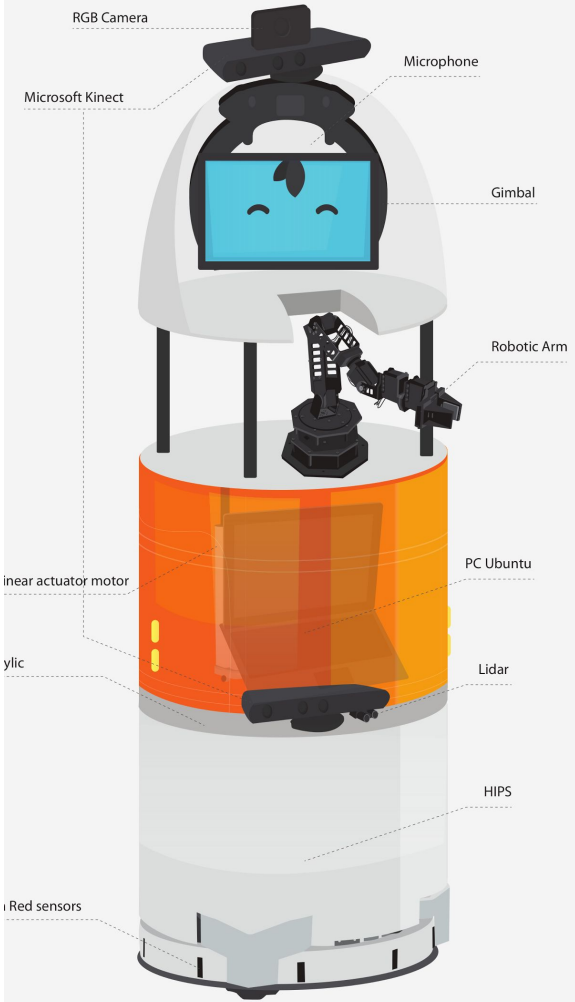
Uma chamada RPC causa a **execução** de uma **Procedure** em **outro endereço** no espaço.

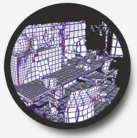
CLIENTE SERVER INTERACTION

INTER PROCESS COMMUNICATION




Publisher
Message
Topic
Subscriber
Subscriber
...
Subscriber
Sender
Message
Queue
Receiver
Receiver
...
Receiver
Requester
Message
Request Channel
Message
Reply Channel
"Method"?
Replier





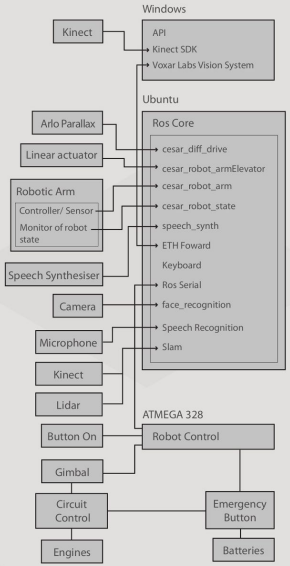
Pattern Recognition
Makes the robot able to understand its environment, detecting and recognizing various types of objects.

Object Tracking
In addition to detect and recognize, the robot can track the movement of objects, locating them spatially in the environment.

Facial Recognition
Detects the position of a face and recognizes the possible expressed emotions such as joy, surprise, fear, sadness ...

People Tracking
Identifies the movement of the user in order to follow him.

System Architecture

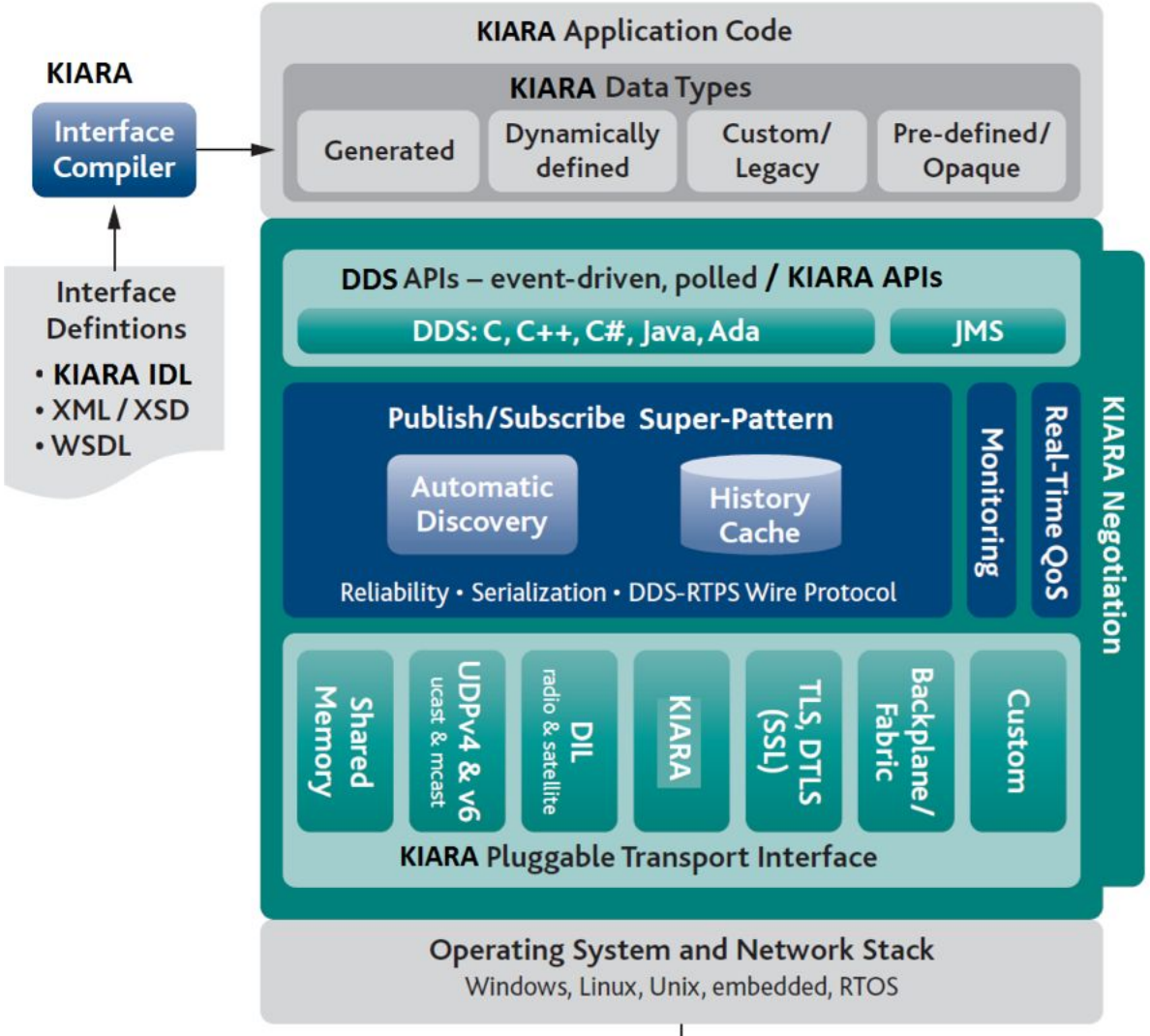


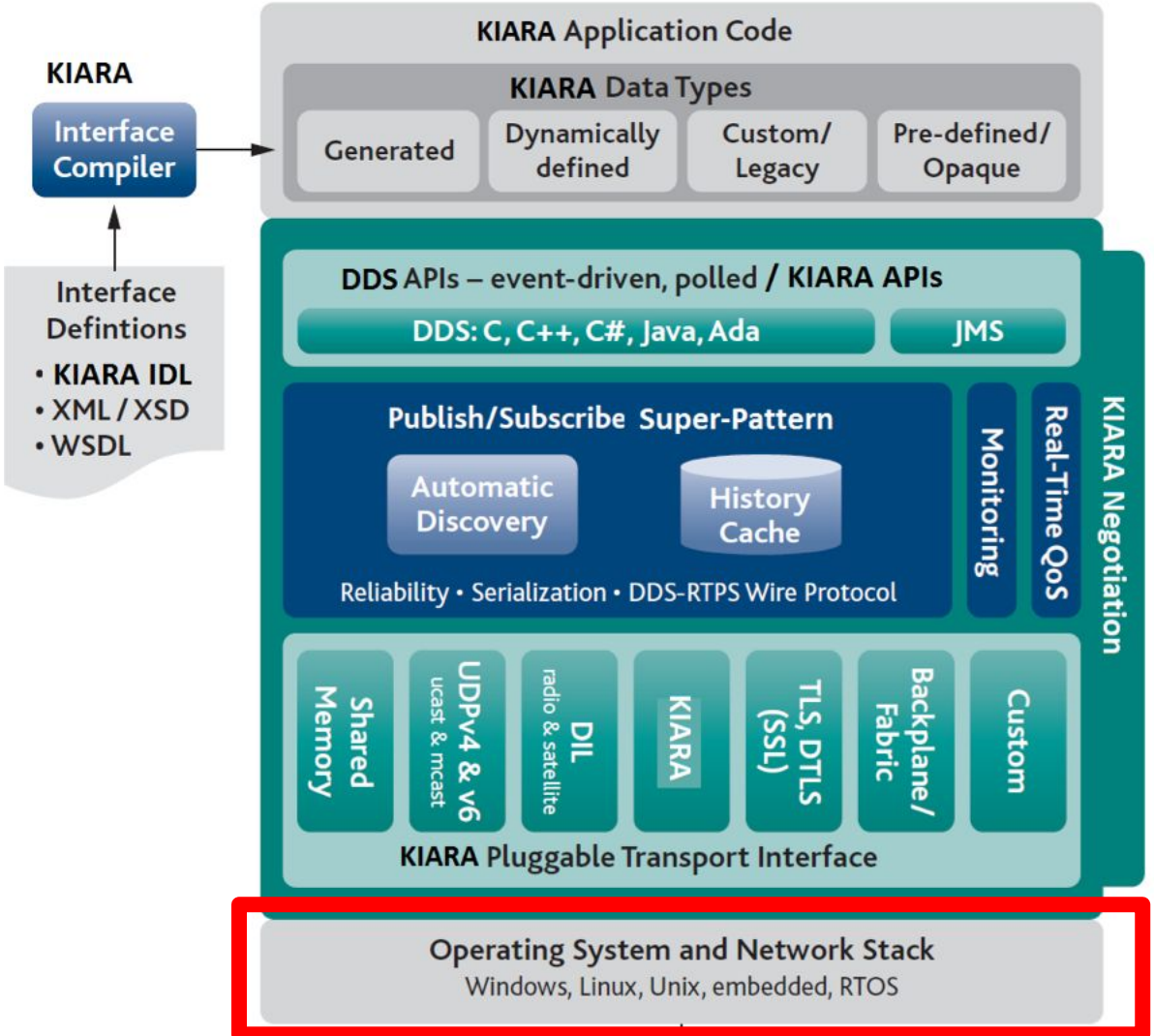
Open specification

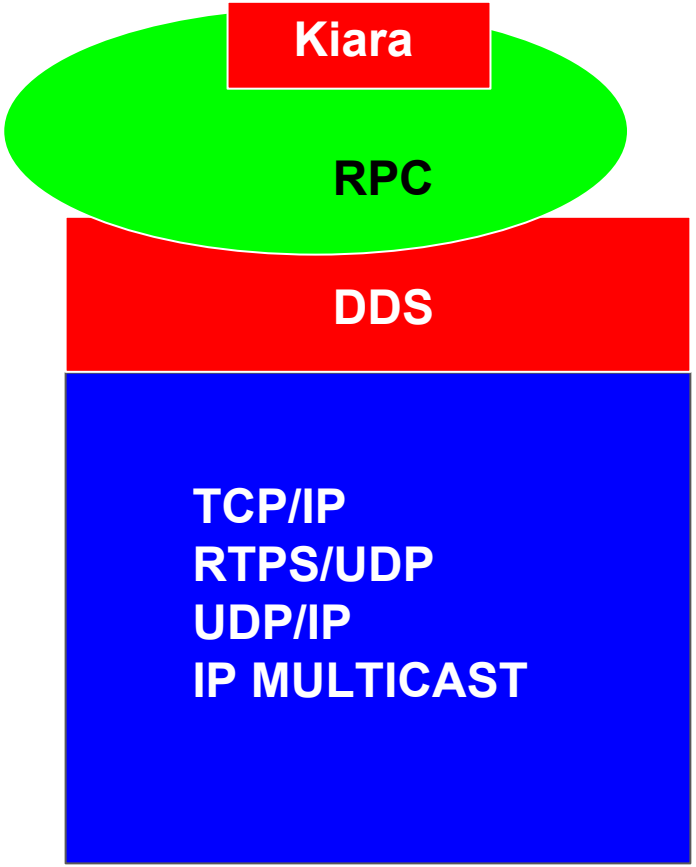
Kiara has been designed according to the **Advanced Middleware Open Specification.**

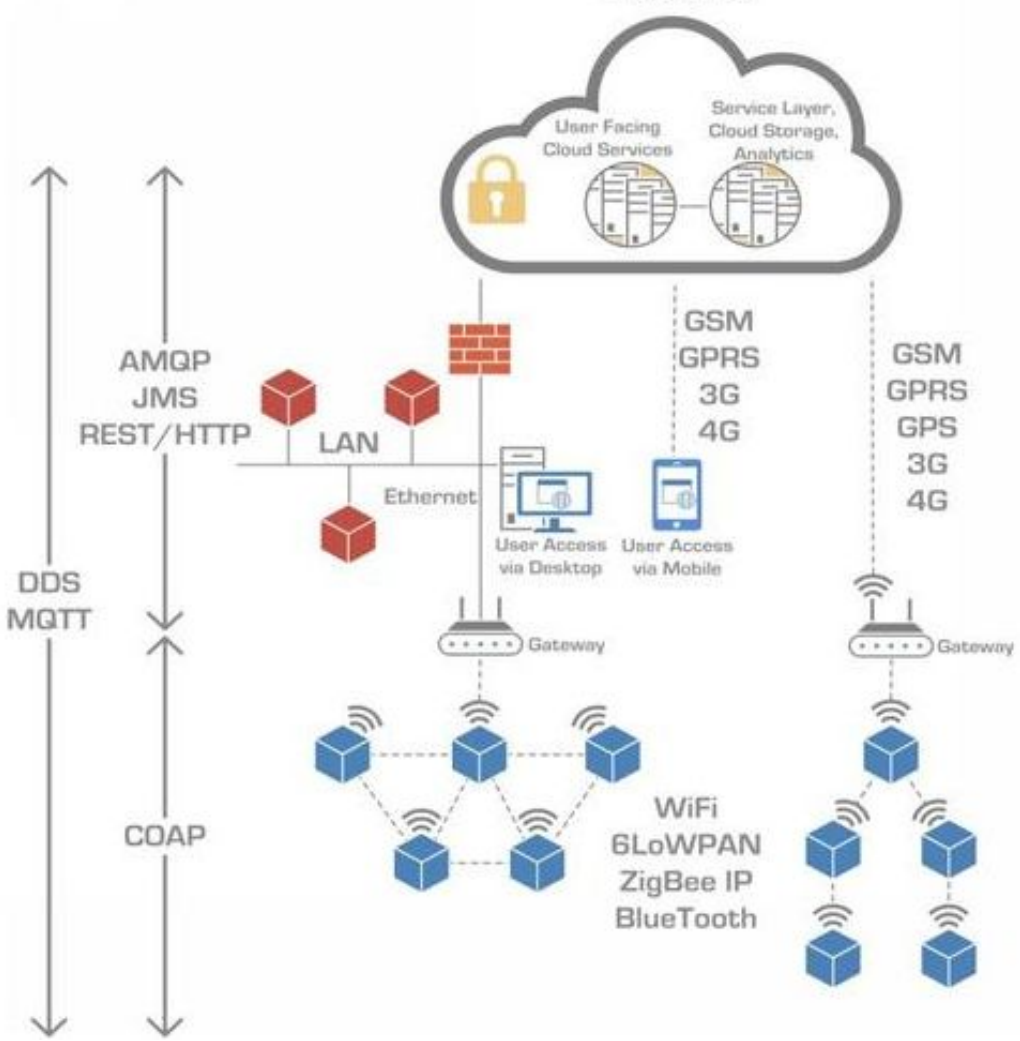
Kiara is an implementation of the **FIWARE Advanced Middleware Generic Enabler. APIs:**

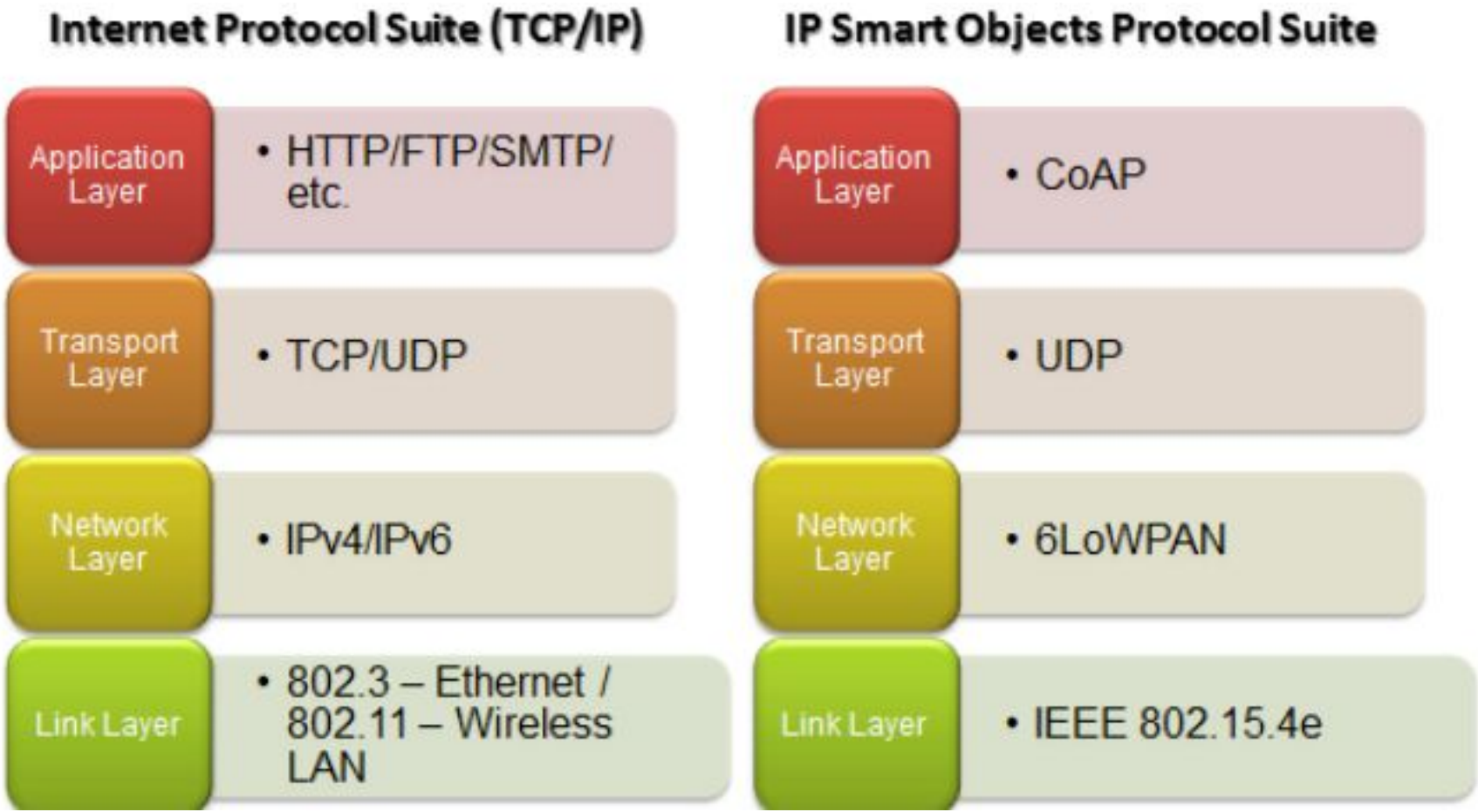
- **Kiara Advanced Middleware Specification**
- **Kiara Advanced Middleware IDL Specification**
- **Kiara Advanced Middleware RPC API Specification**
- **Kiara Advanced Middleware RPC Dynamic Types API**
- **Kiara Advanced Middleware Publish/Subscribe Specification**


KIARA Application Code
KIARA
Interface Compiler
KIARA Data Types
Generated
Dynamically defined
Custom/ Legacy
Pre-defined/ Opaque
Interface Defintions
• KIARA IDL
• XML / XSD
• WSDL
DDS APIs – event-driven, polled / KIARA APIs
DDS: C, C++, C#, Java, Ada
JMS
Publish/Subscribe Super-Pattern
Automatic Discovery
History Cache
Reliability • Serialization • DDS-RTPS Wire Protocol
Monitoring
Real-Time QoS
KIARA Negotiation
Shared Memory
UDPv4 & v6
ucast & mcast
DIL
radio & satellite
KIARA
TLS, DTLS (SSL)
Backplane/ Fabric
Custom
KIARA Pluggable Transport Interface
Operating System and Network Stack
Windows, Linux, Unix, embedded, RTOS


KIARA
Interface Compiler
Interface Defintions
• KIARA IDL
• XML / XSD
• WSDL
KIARA Application Code
KIARA Data Types
Generated
Dynamically defined
Custom/ Legacy
Pre-defined/ Opaque
DDS APIs – event-driven, polled / KIARA APIs
DDS: C, C++, C#, Java, Ada
JMS
Publish/Subscribe Super-Pattern
Automatic Discovery
History Cache
Reliability • Serialization • DDS-RTPS Wire Protocol
Monitoring
Real-Time QoS
KIARA Negotiation
Shared Memory
UDPv4 & v6 ucast & mcast
DIL radio & satellite
KIARA
TLS, DTLS (SSL)
Backplane/ Fabric
Custom
KIARA Pluggable Transport Interface
Operating System and Network Stack
Windows, Linux, Unix, embedded, RTOS


Kiara
RPC
DDS
TCP/IP
RTPS/UDP
UDP/IP
IP MULTICAST


User Facing
Cloud Services
Service Layer,
Cloud Storage,
Analytics
AMQP
JMS
REST/HTTP
LAN
Ethernet
GSM
GPRS
3G
4G
GSM
GPRS
GPS
3G
4G
User Access
via Desktop
User Access
via Mobile
DDS
MQTT
Gateway
Gateway
COAP
WiFi
6LoWPAN
ZigBee IP
BlueTooth


Internet Protocol Suite (TCP/IP)
Application Layer
• HTTP/FTP/SMTP/ etc.
Transport Layer
• TCP/UDP
Network Layer
• IPv4/IPv6
Link Layer
• 802.3 – Ethernet / 802.11 – Wireless LAN
IP Smart Objects Protocol Suite
Application Layer
• CoAP
Transport Layer
• UDP
Network Layer
• 6LoWPAN
Link Layer
• IEEE 802.15.4e

	Transport	Paradigm	Scope	Discovery	Content Awareness	Data Centricity	Security	Data Prioritisation	Fault Tolerance
AMQP	TCP/IP	Point-to-Point Message Exchange	D2D D2C C2C	No	None	Encoding	TLS	None	Impl. Specific
CoAP	UDP/IP	Request/ Reply (REST)	D2D	Yes	None	Encoding	DTLS	None	Decentrailsed
DDS	UDP/IP (unicast + mcast) TCP/IP	Publish/ Subscribe Request/ Reply	D2D D2C C2C	Yes	Content-Based Routing, Queries	Encoding, Declaration	TLS, DTLS, DDS Security	Tranport Priorities	Decentralised
MQTT	TCP/IP	Publish/ Subscribe	D2C	No	None	Undefined	TLS	None	Broker is the SPoF

Protocol	CoAP	XMPP	RESTful HTTP	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G Suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Stories	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management,home services)	Extending enterprise messaging into IoT applications

A QUICK EXAMPLE

In the IDL derived approach, first the IDL definition has to be created:

```
service Calculator {
    float32 add (float32 n1, float32 n2);
    float32 subtract (float32 n1, float32 n2);
};
```

The developer has to implement the functions inside the class CalculatorServantImpl:

```
public static class CalculatorServantImpl extends CalculatorServant
{
    @Override
    public float add (/*in*/ float n1, /*in*/ float n2) {
        return (float) n1   n2;
    }

    @Override
    public float subtract (/*in*/ float n1, /*in*/ float n2) {
        return (float) n1 - n2;
    }
```

Now the server can be started:

```
// Create context, server and service
Context context = Kiara.createContext();
Server server = context.createServer();
Service service = context.createService();

// Create and register an instance of the CalculatorServant implementation.
CalculatorServant Calculator_impl =new CalculatorServantImpl();
service.register(Calculator_impl);

// register the service on port 9090 using CDR serialization
server.addService(service, "tcp://0.0.0.0:9090", "cdr");

// run the server
server.run();
```

The client can connect and call the remote functions via the proxy class:

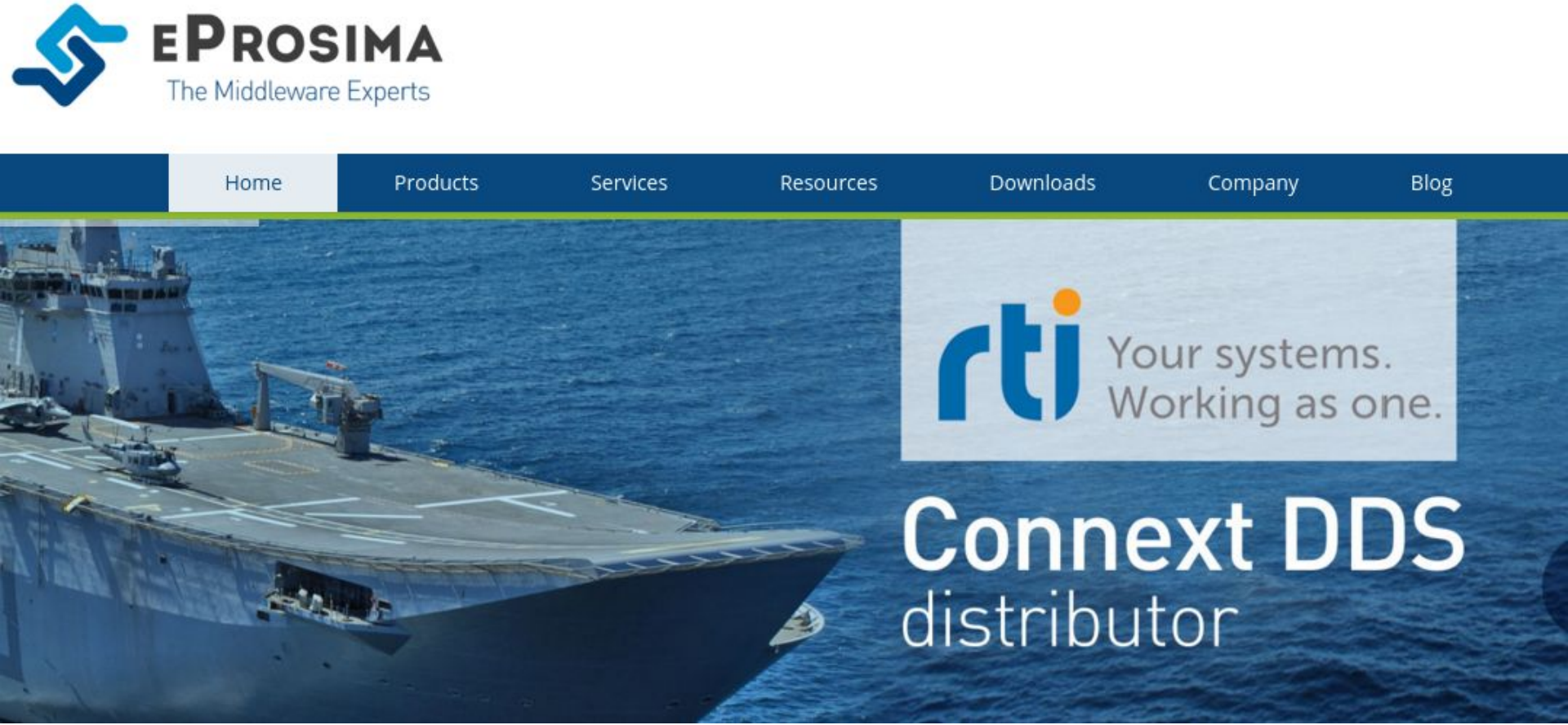
```
// Create context
Context context = Kiara.createContext();

// setup the connection to the server
Connection connection = context.connect"tcp://192.168.1.18:9090?serialization=cdr);

// get the client Proxy implementation
CalculatorClient client = connection.getServiceProxy(CalculatorClien class);

// Call the remote methods
float result = client.add 3, 5);
```

PRIMEIRO FUNDAMENTO


eProsima
The Middleware Experts
Home
Products
Services
Resources
Downloads
Company
Blog
rti
Your systems.
Working as one.
Connext DDS
distributor

eProsima and the Robot Operating System (ROS)

eProsima is a contributor of ROS, the Robot Operating System, providing networking middleware for the upcoming release. Our product eProsima Fast RTPS has been selected as one of the middleware options available in ROS 2.

eProsima Fast RTPS is an open source product, and it is available on GitHub and our website free of charge. As a result, several important robotics companies are using or evaluating eProsima Fast RTPS for their Robots.

FIWARE Advanced Middleware - ROS Enabled.

eProsima is the middleware leader for the FIWARE (Future Internet WARE) European initiative developing the FIWARE Advanced Middleware. The FIWARE European initiative has created a set of software technologies and tools freely available for the European Startups and entrepreneurs.

KIARA Includes a Java implementation of eProsima Fast RTPS, ensuring interoperability with ROS applications.

The eProsima Advantage

RTPS was designed as a protocol for Robotics, and our products implement this protocol both in Java and C/C , as part of the company strategy to offer open source solutions to this sector.


ROS

3Dr USA eProsima
http://diydrones.com/profiles/blogs/3d-robotics-selects-eprosima-fast-rtps-for-system-infrastructure

DDS com Rasp
http://rticommunity.github.io/rticonnextdds-android-raspberrypi-demo/

https://youtu.be/lWMZbyZvkRs?t=328

WWW.I-ZAK.ORG