

# MQX Application Builder

Doug Bruce

Embedded Access Inc

[Doug.Bruce@NXP.com](mailto:Doug.Bruce@NXP.com)

September 2019 | Session #AMF-SOL-T3596



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# Agenda

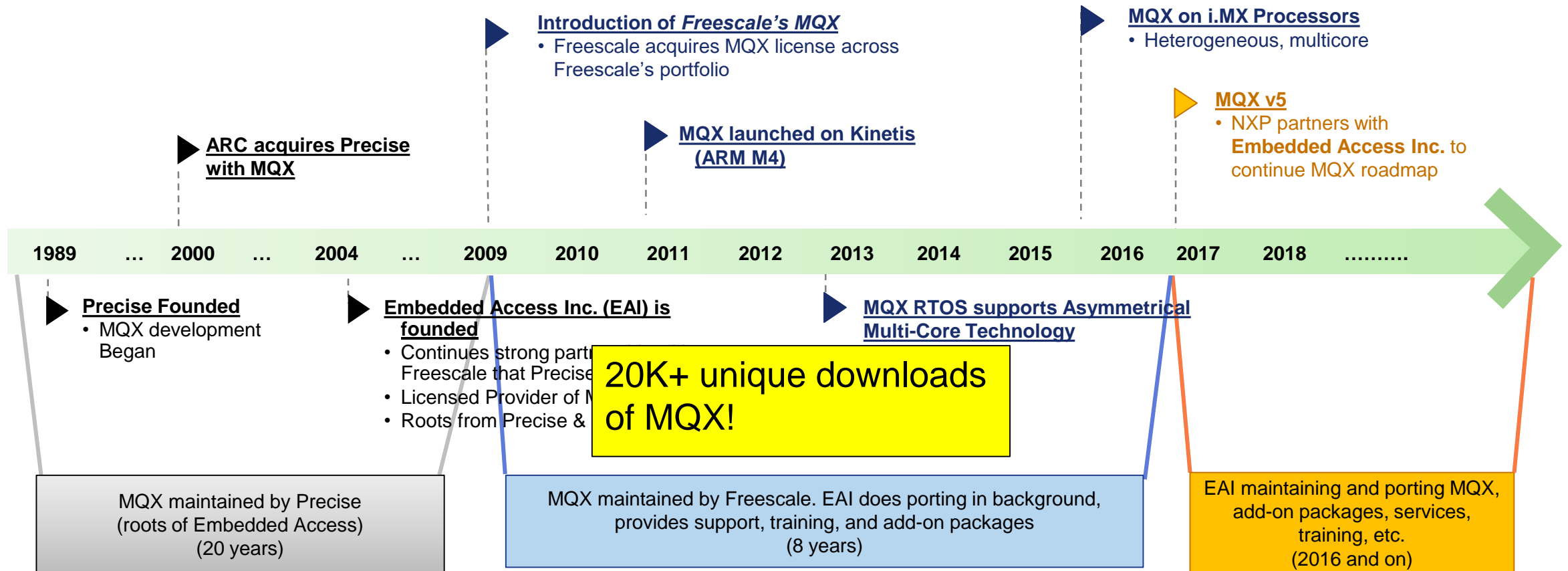
---

- i.MX RT Series Summary
- Background on MQX and EAI
- The Typical Method to Creating a Networking Application
- What is the MQX Application Builder?
- Advantages
- How It Works
- Hands-on Lab

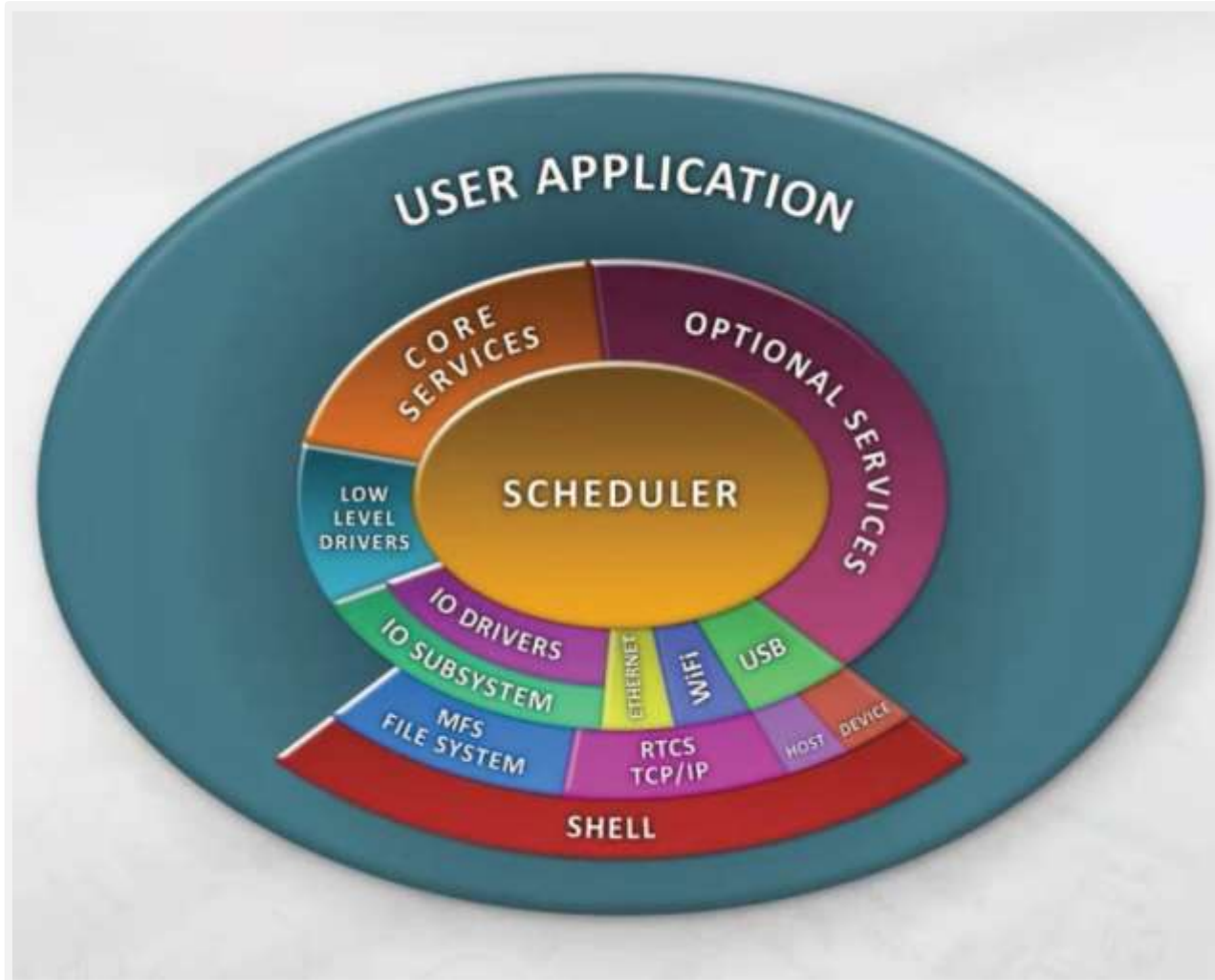
# Background to MQX



# MQX Timeline



# What is MQX



## Additions in MQX 5.x:

- IoT protocols: MQTT and REST
- IPv6 compliant protocols
- lightweight JSON Parser, cJSON Parser / Framer
- lightweight XML Parser / Framer
- zlib Compression / Decompression utility
- RPMsg for inter-core communication
- Lua Scripting Language
- Fixes for identified security vulnerabilities
- Improved robustness
- many MISRA rules supported
- C99 standard compliance
- Driver Enhancements
- Bug Fixes
- New processor ports
- Etc.

# Where is MQX Used?

## Industrial Control

- Event Recorder
- Power Monitoring
- Weigh Scale Equipment
- Solar Panel Inverter
- Oil Flow Monitoring

## Automotive

- Car Radio
- CAN Gateway
- Data Bridge
- Vehicle Subsystem Controller

## Transportation

- GSM Data Comm Hub
- Breath Analyser and Camera
- Marine Satellite Tracking
- Railway Signalling

## Navigation

- GPS Tracking System
- Military Navigation System
- Geofence Monitoring

## Medical

- Drug Injection unit
- Proton Therapy dose monitoring
- Laser eye surgery equip.

## Test & Measurement

- Spectrum Analyzer
- Power Supply
- Seismic Monitor
- Power Measurement

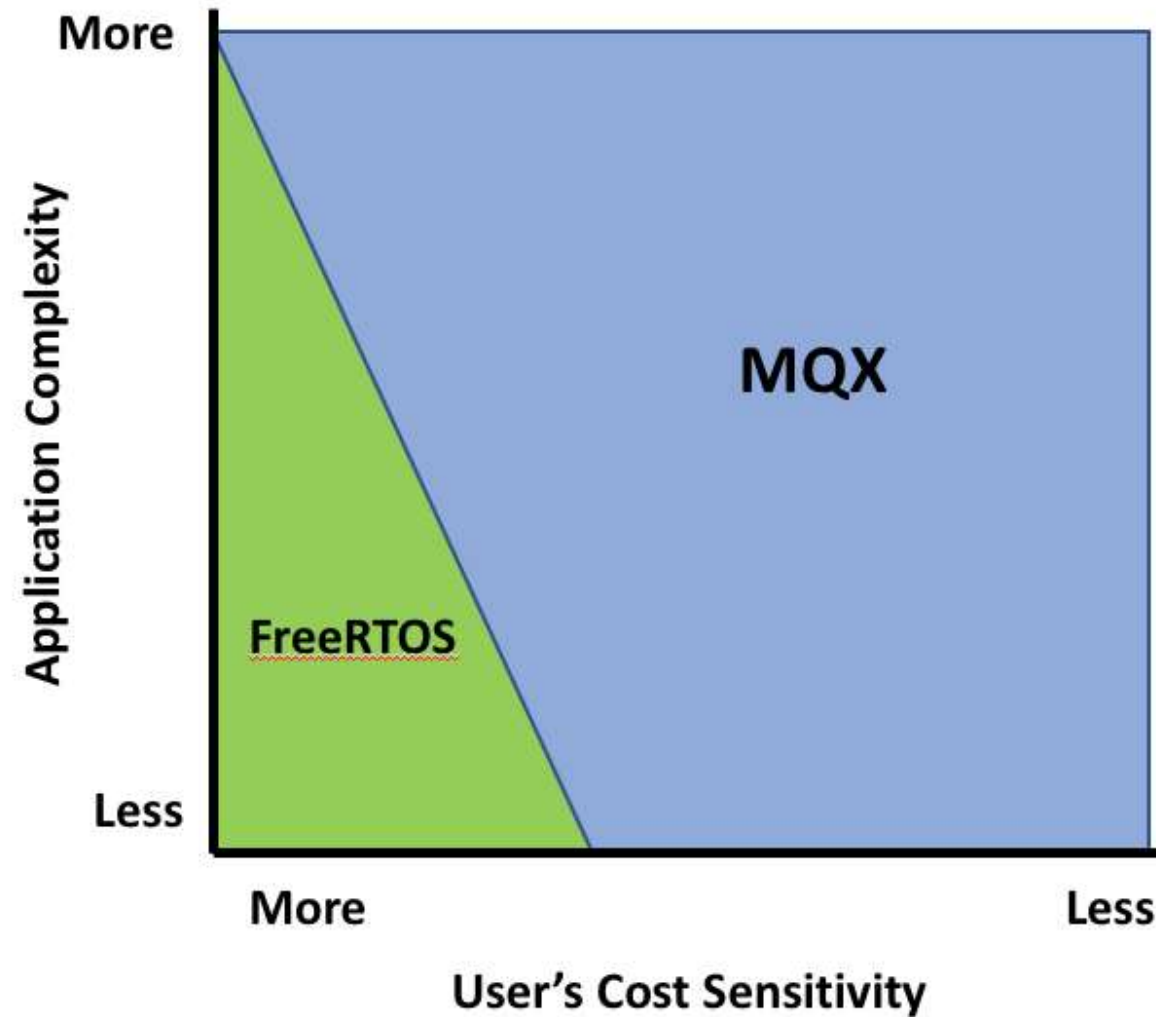
## Telecom

- Optical Switching Unit
- Base Station Transmission

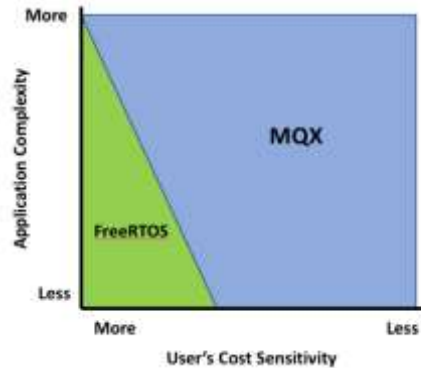
## Consumer

- Wireless Blinds Controller
- Shower Spa Controller
- Wireless Dog Collar

# MQX vs FreeRTOS



# MQX vs FreeRTOS



FreeRTOS is adequate for some applications but:

1. FreeRTOS is just a micro kernel, it not really an RTOS. The vast majority of features available in MQX are missing in FreeRTOS
2. It is owned by Amazon which poses a risk
3. It offers no useful user documentation
4. It is a self taught course. MQX has a set of 20 training videos
5. FreeRTOS does not support multi-processors at all
6. Inter-core communication is very basic (ping pong messages) in FreeRTOS
7. MQX has extra features above and beyond the SDK
8. MQX has a cleaner, more consistent API
9. MQX offers a simpler environment to create an application from scratch. Now with the MQX App Builder that extends to all of the networking aspects.
10. FreeRTOS offers NO differentiation!

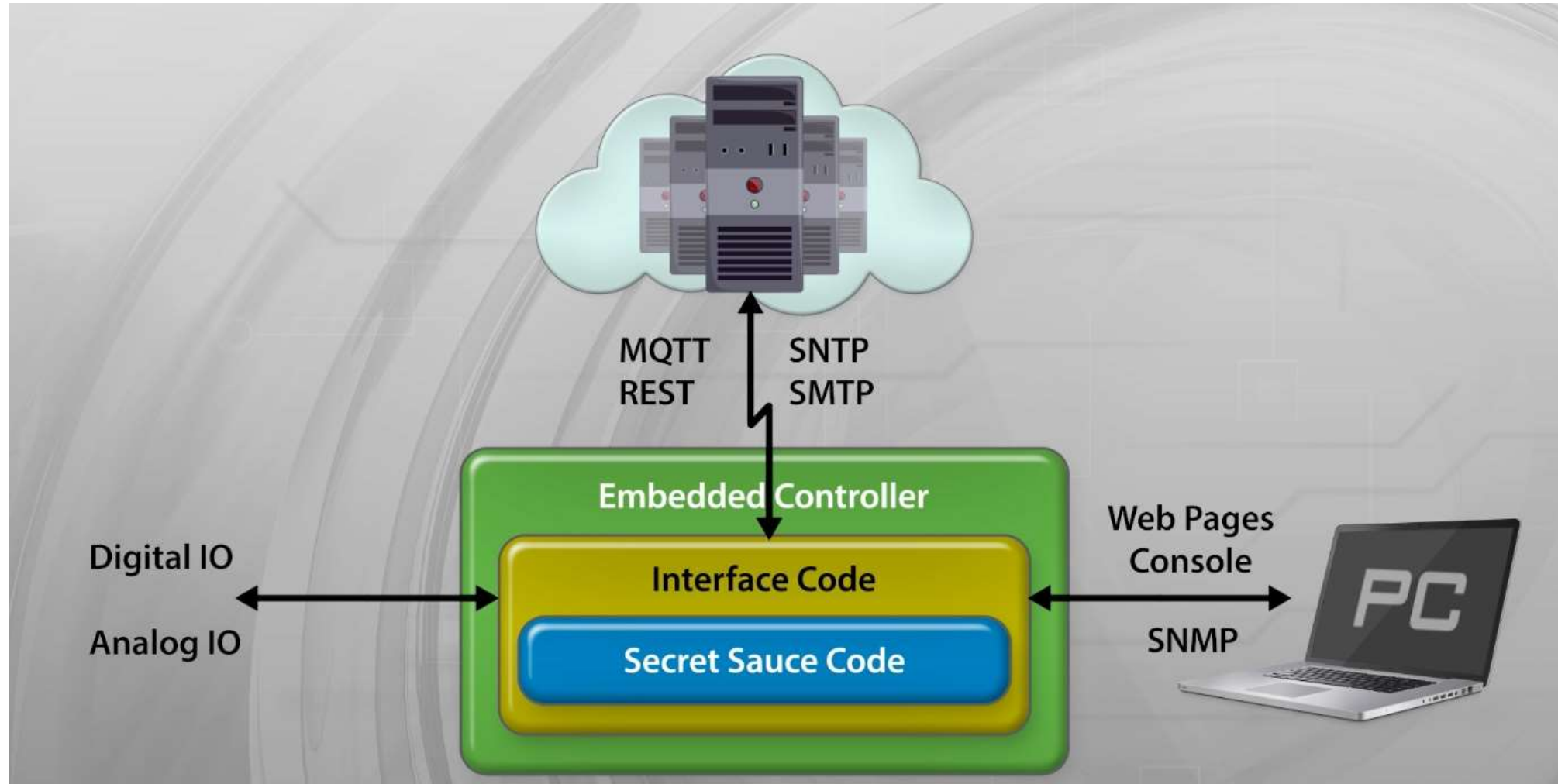


# How Networking Applications are Traditionally Created

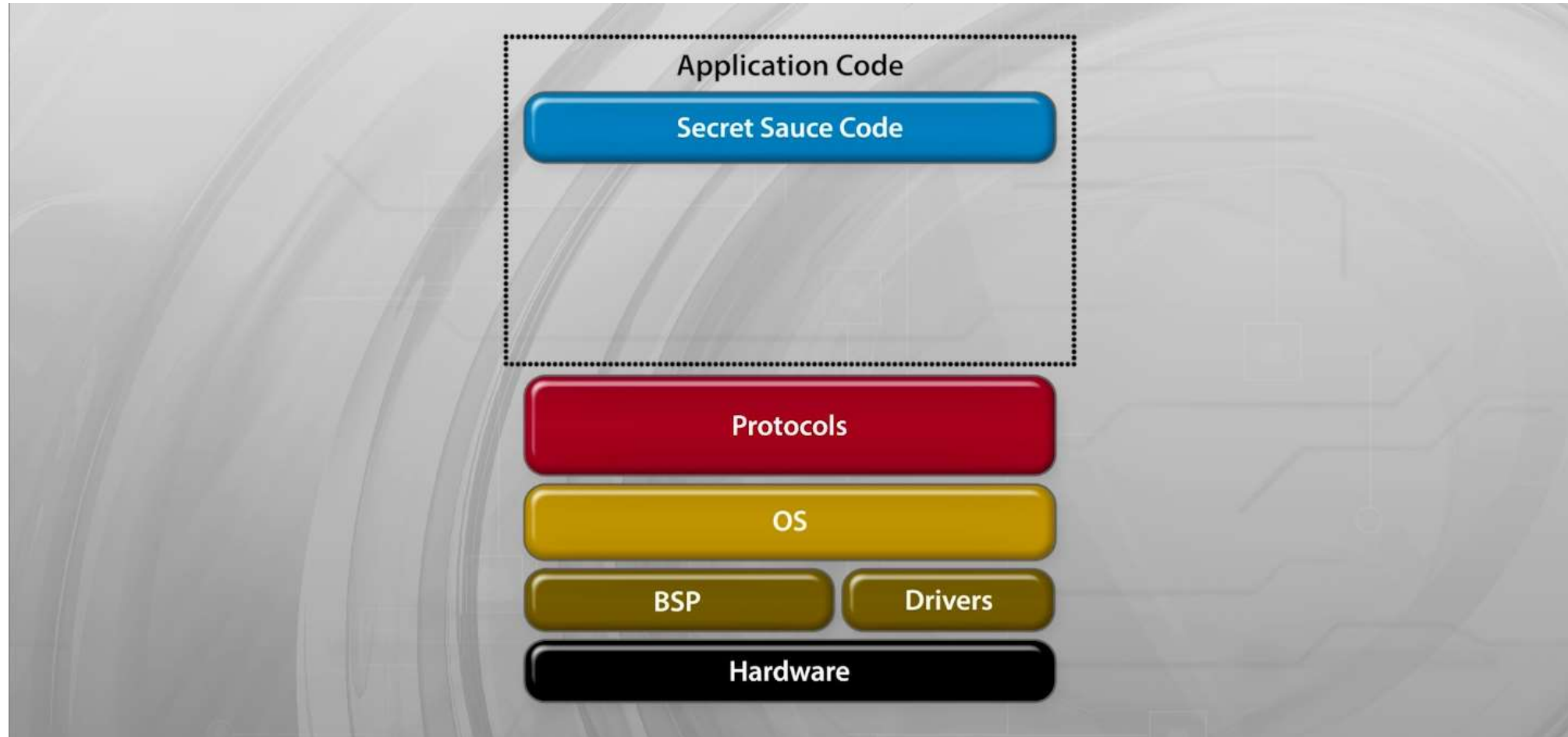
The turn the crank, brute force approach



# Typical Application



# Typical Architecture



# What's Traditionally Involved in Creating an Application

1. Start with your “secret sauce” application
2. Port and Configure protocol stack(s)
3. Add device interfaces
4. Add management of device interfaces
5. If web server is used,
  - Write CGI and ASP handling code
  - Write Web pages
6. If SNMP is used
  - Write MIB
  - Write MIB Handler
  - Write Trap generators
7. If RESTful Server API is used
  - Modify HTTP server to support REST
  - Write Rest handlers
  - Port or write XML or JSON parser framers

# What's Traditionally Involved in Creating an Application

8. If RESTful Client API is used:
  - Write REST Client
  - Integrate SSL Client if needed
9. If telnet or shell is used:
  - Write application specific shell commands
10. If SSH is needed
  - Port SSH,
  - Add SSH to Shell interface
11. If email notification is needed:
  - Integrate SMTP / TLS
  - Write code to generate emails
12. If MQTT is needed:
  - Port MQTT Stack
  - write MQTT subscription and publication functions

And so on for each  
Internet protocol  
required...

# What's Traditionally Involved in Creating an Application

13. And then there is the housekeeping stuff:

- Provide some sort of access control
- Mount file systems
- Store and retrieve configuration data
- Bootloader
- Upload new image
- Upload/download files
- Logging

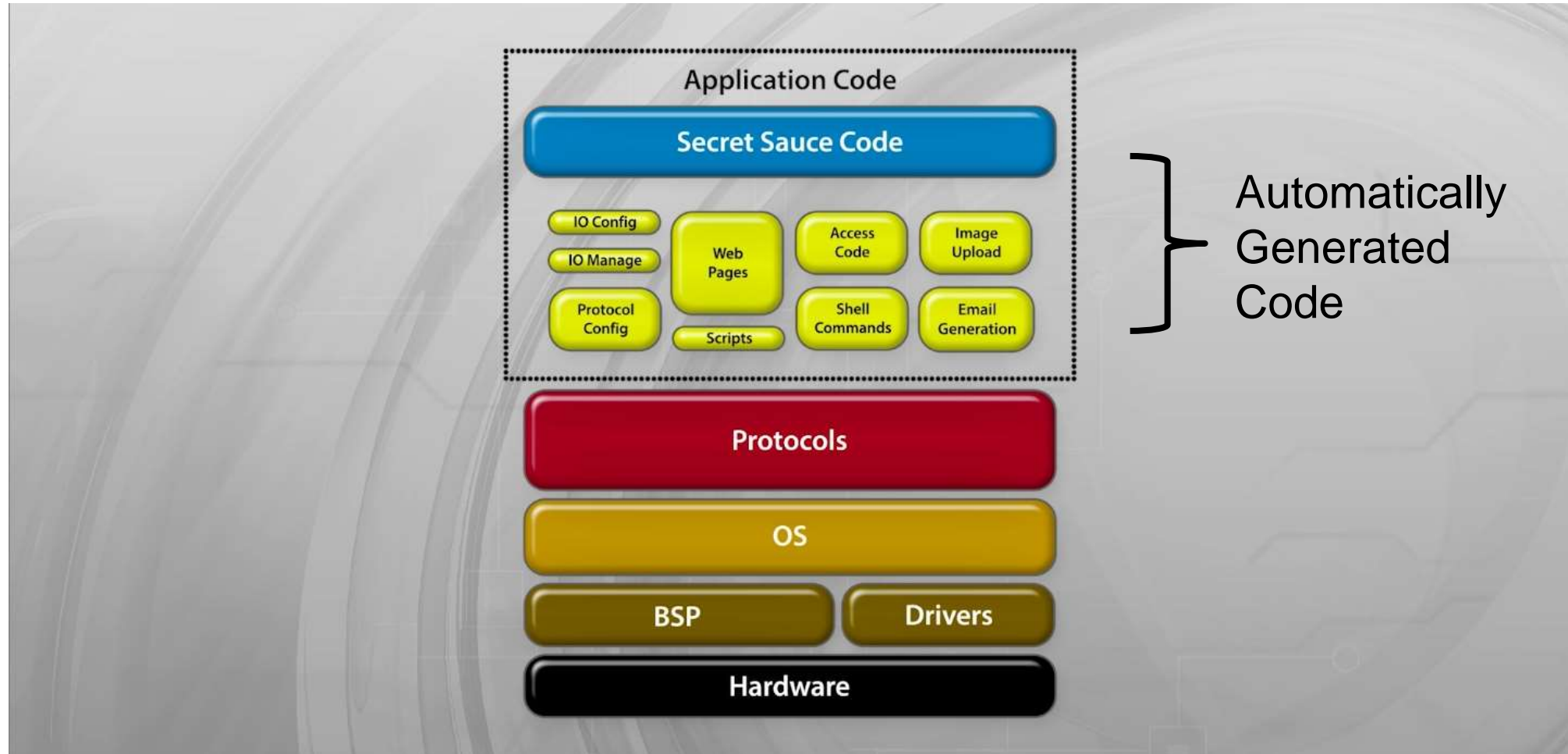
# How do Most MQX Users Start?

## **web\_hvac app that comes with MQX**

What's wrong with that? And we can criticize because we wrote it...

- Each CGI is hand crafted for a specific webpage
  - Add a webpage, add a CGI
  - Change a webpage, change a CGI
- Each Shell command is hand crafted
- Lots of code for someone to wade through, much of it is demo specific
- And it only supports HTTP and telnet

# Typical Architecture



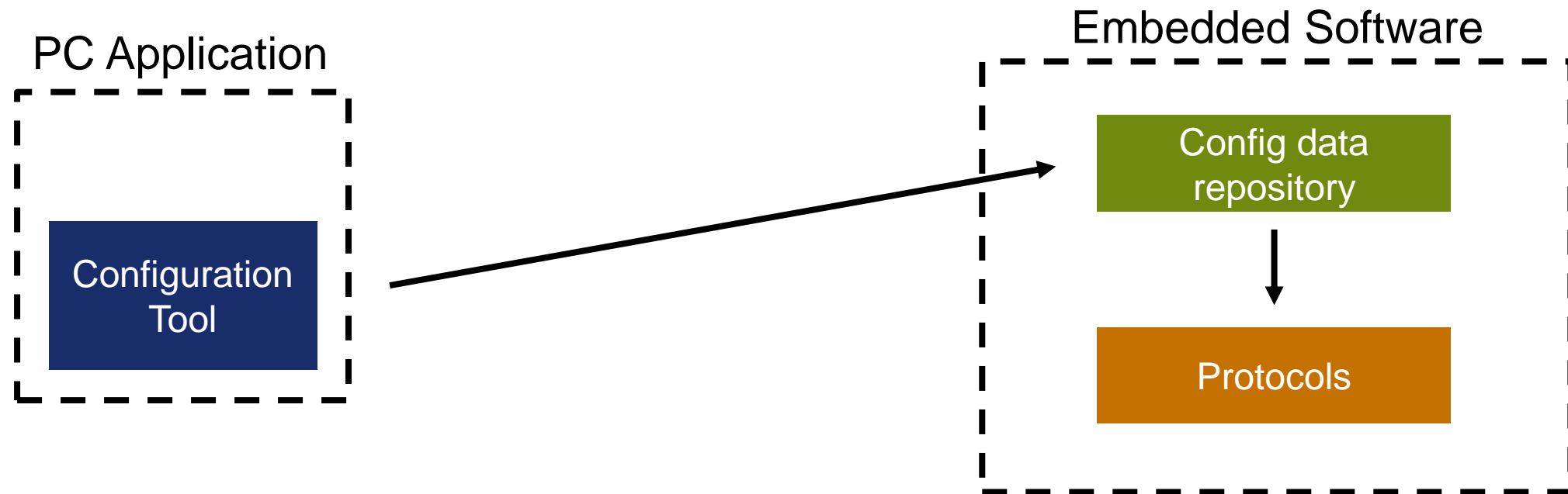


# A Better Way – The MQX Application Builder



## A Better Way

1. Configure what you want in a user interface that exports all entered data to a central repository
2. In MQX, for each protocol
  - Provide protocol implementation
  - Provide Protocol Interface Layer that populates required data from central repository



# MQX Application Builder: What It Is

An extension to the MCUXpresso Config Tools that enables you to define all key components of a networking application included the required protocols

Screen selection

Generate code

Data entry

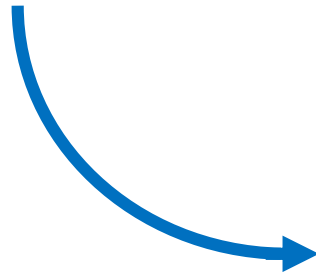
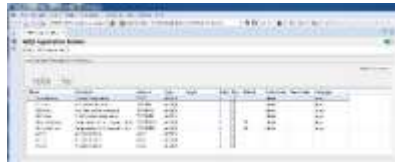
The screenshot shows the MQX Application Builder interface. The 'App' tab is selected, and the 'Update Code' button is highlighted. The table below contains the following data:

Name	Description	Acronym	Type	Length	Index	Sto...	Default	Write Access	Read Access	Web page
CurrentTemp	Current Temperature	TEMP	uint32_t		1	<input type="checkbox"/>		admin		home
SetTemp	Desired Temperature	TEMPSET	uint32_t		1	<input type="checkbox"/>		admin		home
MaxTemp	Max Recorded temperature	TEMPMAX	uint32_t		1	<input type="checkbox"/>		admin		home
MinTemp	Min Recorded temperature	TEMPMIN	uint32_t		1	<input type="checkbox"/>		admin		home
MaxTempAlarm	Temperature Alarm set point - high	TEMPMAXA	uint32_t		1	<input type="checkbox"/>	25	admin		home
MinTempAlarm	Temperature Alarm set point - low	TEMPMINA	uint32_t		1	<input type="checkbox"/>	18	admin		home
AccelX	Accelerometer X	ACCX	uint32_t		1	<input type="checkbox"/>				
AccelY	Accelerometer Y	ACCY	uint32_t		1	<input type="checkbox"/>				
AccelZ	Accelerometer Z	ACCZ	uint32_t		1	<input type="checkbox"/>				

# MQX Application Builder in 4 Steps: Step 1

The easy way to configure your application and networking protocols

## Step 1



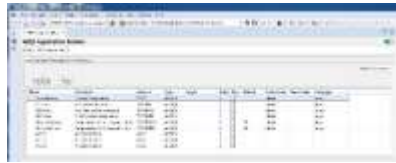
### Configure:

- Key application variables
- Interfaces to use (ethernet, wifi, etc)
- Protocols: MQTT, SSL, SNMP, SNTP, Rest, email, etc
- Digital / Analog IO used
- Alarms, logging, user access levels

# MQX Application Builder in 4 Steps: Step 2

The easy way to configure your application and networking protocols

Step 1



Step 2



Click on 'Update Code'



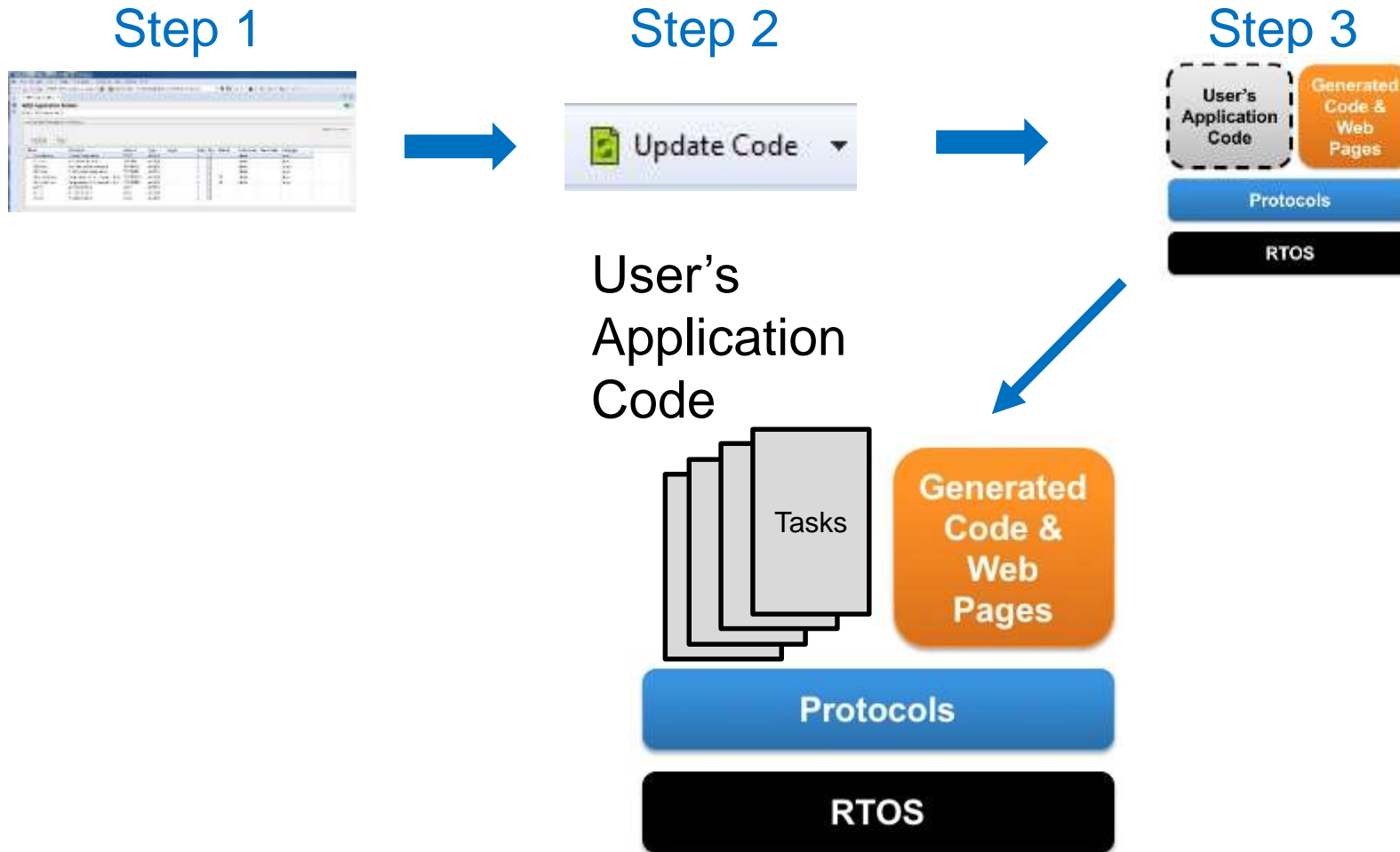
Web pages

MetaData

This generates the web pages  
and metadata files

# MQX Application Builder in 4 Steps: Step 3

The easy way to configure your application and networking protocols



# MQX Application Builder in 4 Steps: Step 4

The easy way to configure your application and networking protocols



# Key Advantages

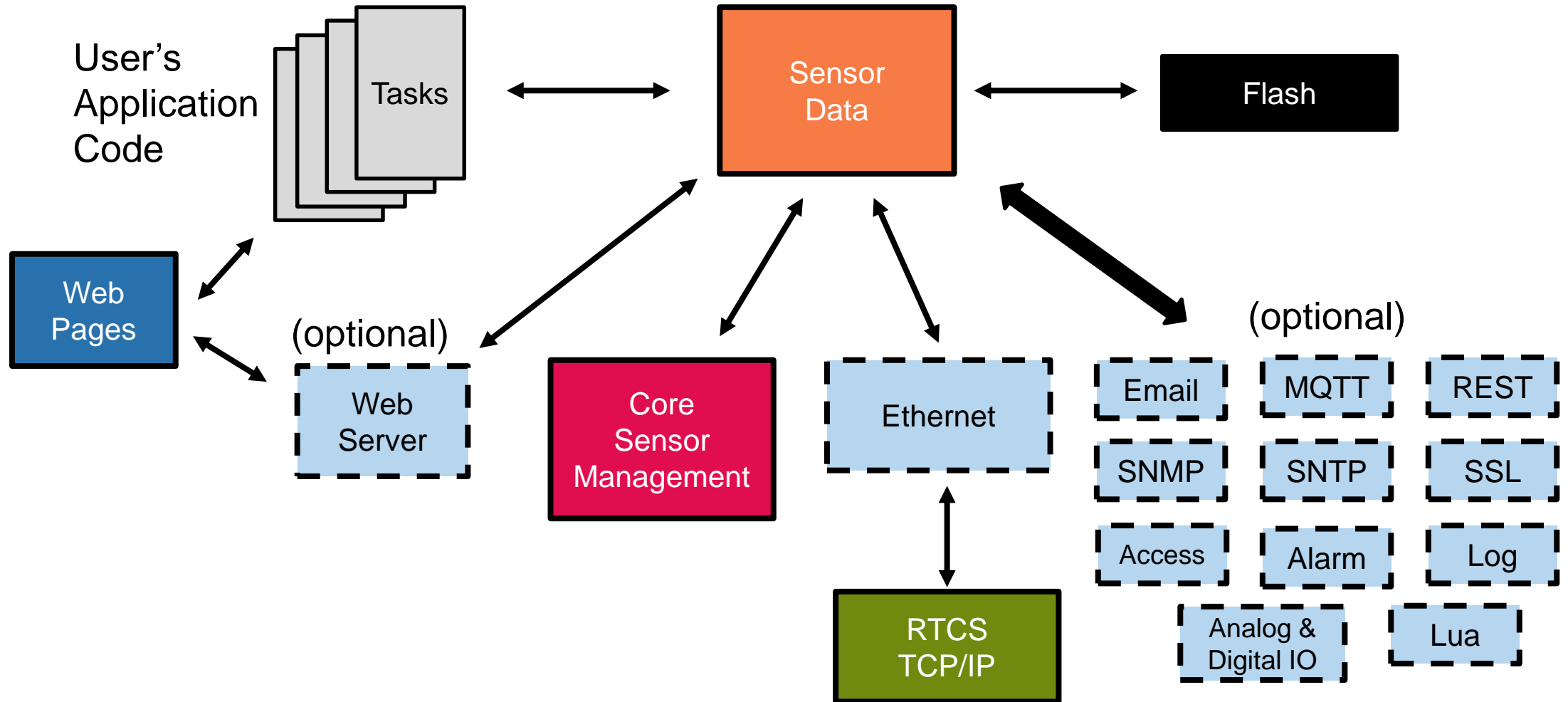
- Allows customers to focus on application layer
- Rapid migration of existing products and prototyping of new products
- Getting started with any OS, including MQX, has its challenges and the App Builder provides a huge boost to the prototyping stage
- Many users don't want to spend the time to learn the Networking Protocols
- Easy way to demo a new application or a port of an existing application



# How it Works



# Accessing the Generated Code



# Accessing Application and Internal Sensor Data

Your application code can access and update any of the internal variables using a collection of 'Get' and 'Set' commands.

## Getting a Value:

Sensor\_get\_boolean()  
Sensor\_get\_uint32\_t()  
Sensor\_get\_type()  
Sensor\_get\_string()  
Sensor\_get\_mac\_address()  
Sensor\_get\_read\_access()

- 
- 
- 

## Setting a Value:

Sensor\_set\_string()  
Sensor\_set\_int32\_t()  
Sensor\_set\_double()  
Sensor\_set\_float()  
Sensor\_set\_ip\_address()  
Sensor\_set\_from\_string()

- 
- 
-

# Hands-on Lab Walkthrough



# Overview – Web HVAC Demo the Old Way

Essentially, the core of this application is simple, the complexity is in what's around it

```
WHILE
```

```
    Read_ambient_temp
```

```
    IF (HEAT_MODE && temp < desired_temp)
```

```
        Turn on heat and fan
```

```
        Turn off cooling
```

```
    ELSE IF (COOLING_MODE && temp > desired_temp)
```

```
        Turn on cooling and fan
```

```
        Turn off heat
```

```
    ELSE
```

```
        Turn off heat, cool, fan
```

```
END WHILE
```

# Overview – Web HVAC Demo the Old Way

Creating the full application required you to:

Write web pages code

Create CGI scripts

Write app specific shell commands

Write email code

Manage the IO

```
WHILE
  Read_ambient_temp
  IF (HEAT_MODE && temp < target_temp)
    Turn on heat and fan
    Turn off cooling
  ELSE IF (COOLING_MODE && temp > target_temp)
    Turn on cooling and fan
    Turn off heat
  ELSE
    Turn off heat, cool, fan
  END WHILE
```

Write access code

Write boot loader

Configure Analog & digital IO

Configure protocols

Write image upload code

# Using the MQX App Builder – Step 1

We will do this using the App Builder instead to build the basic application

HVAC Application  
Running on the board



Access the application using  
a console connected over a  
serial port



# Using the MQX App Builder – Step 2

Adding an ethernet interface and serving web pages



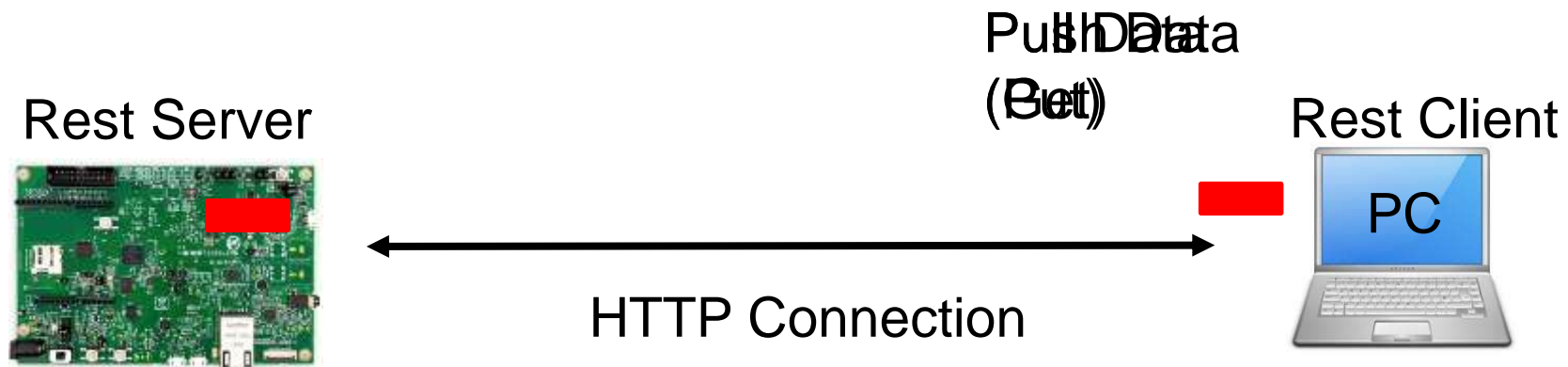
Serve web  
pages





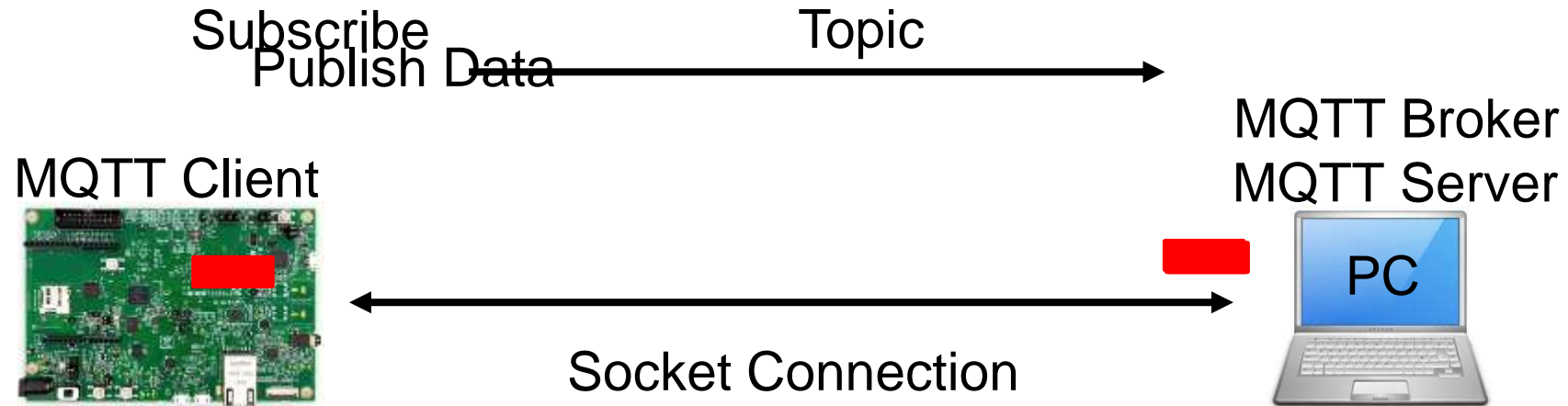
# Using the MQX App Builder – Step 3

Use RESTful API: Representational State Transfer



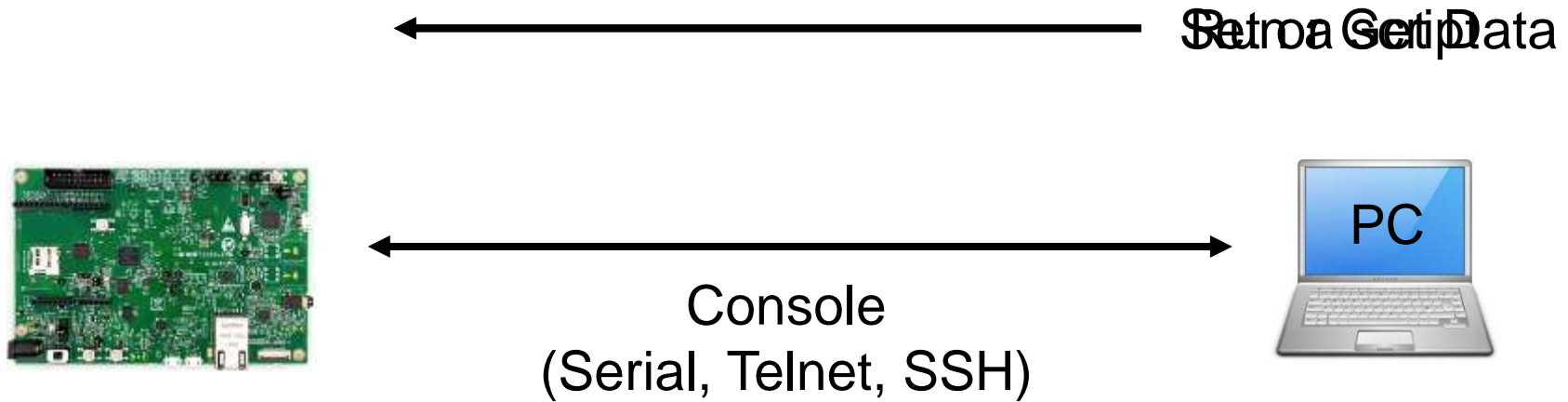
# Using the MQX App Builder – Step 4

Use MQTT: Message Queuing Telemetry Transport

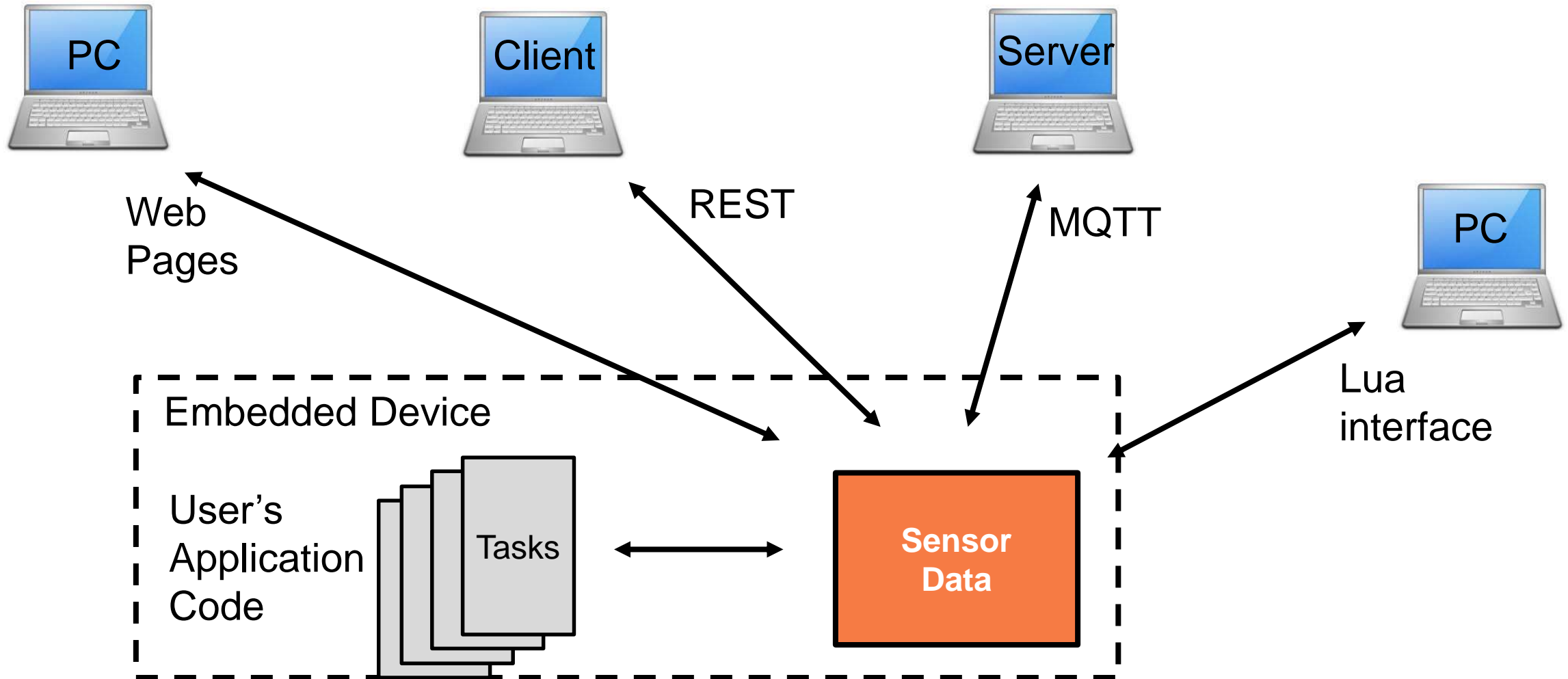


# Using the MQX App Builder – Step 5

Use Lua to remote access internal data or to run an application



# Lab Summary



# References and Contacts

**Session #AMF-SOL-T3596**

[www.nxp.com/mqxappbuilder](http://www.nxp.com/mqxappbuilder)

[www.nxp.com/mqxv5](http://www.nxp.com/mqxv5)

[Doug.Bruce@nxp.com](mailto:Doug.Bruce@nxp.com)

[mqxsales@nxp.com](mailto:mqxsales@nxp.com)

Evaluations coming soon



**SECURE CONNECTIONS  
FOR A SMARTER WORLD**