# The WURFL Framework

Dino Esposito
http://software2cents.wordpress.com
@despos

# **Key Points**

**WURFL** as a DDR

Supported Platforms

**WURFL** Services

# What's WURFL?

## Wireless Universal Resource FiLe

- DDR
- Owned by *ScientiaMobile, Inc.* (**http://www.scientiamobile.com**)

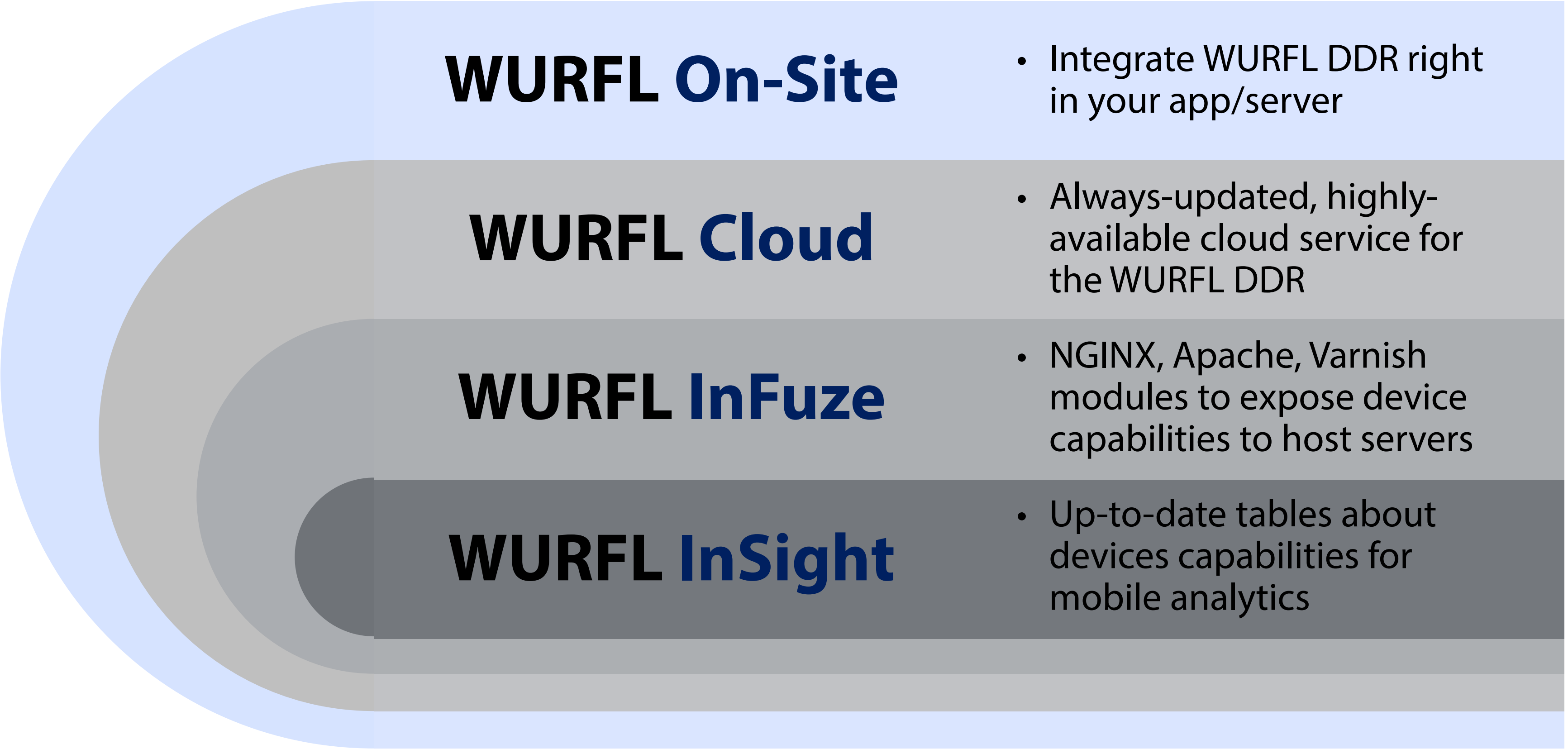
## Multi-language  API

- Java, ASP.NET, C++, PHP, Ruby, Perl, Node.js
- On-premise and cloud

## Structure of the API

- Initialize the DDR
- Pass a user agent string
- Receive a device description with 500+ capabilities

# The WURFL Ecosystem



**WURFL On-Site**

- Integrate WURFL DDR right in your app/server

**WURFL Cloud**

- Always-updated, highly-available cloud service for the WURFL DDR

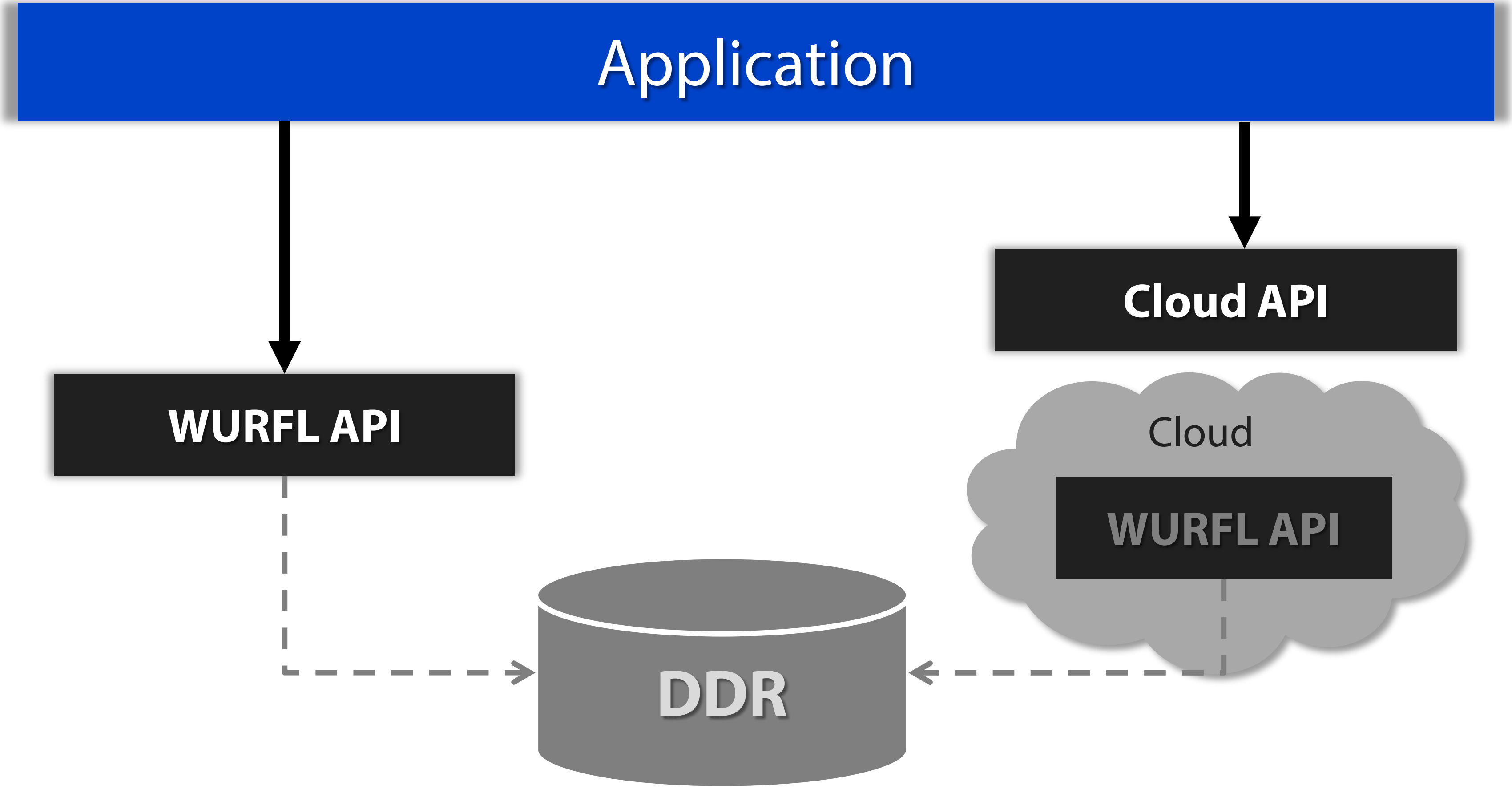
**WURFL InFuze**

- NGINX, Apache, Varnish modules to expose device capabilities to host servers

**WURFL InSight**

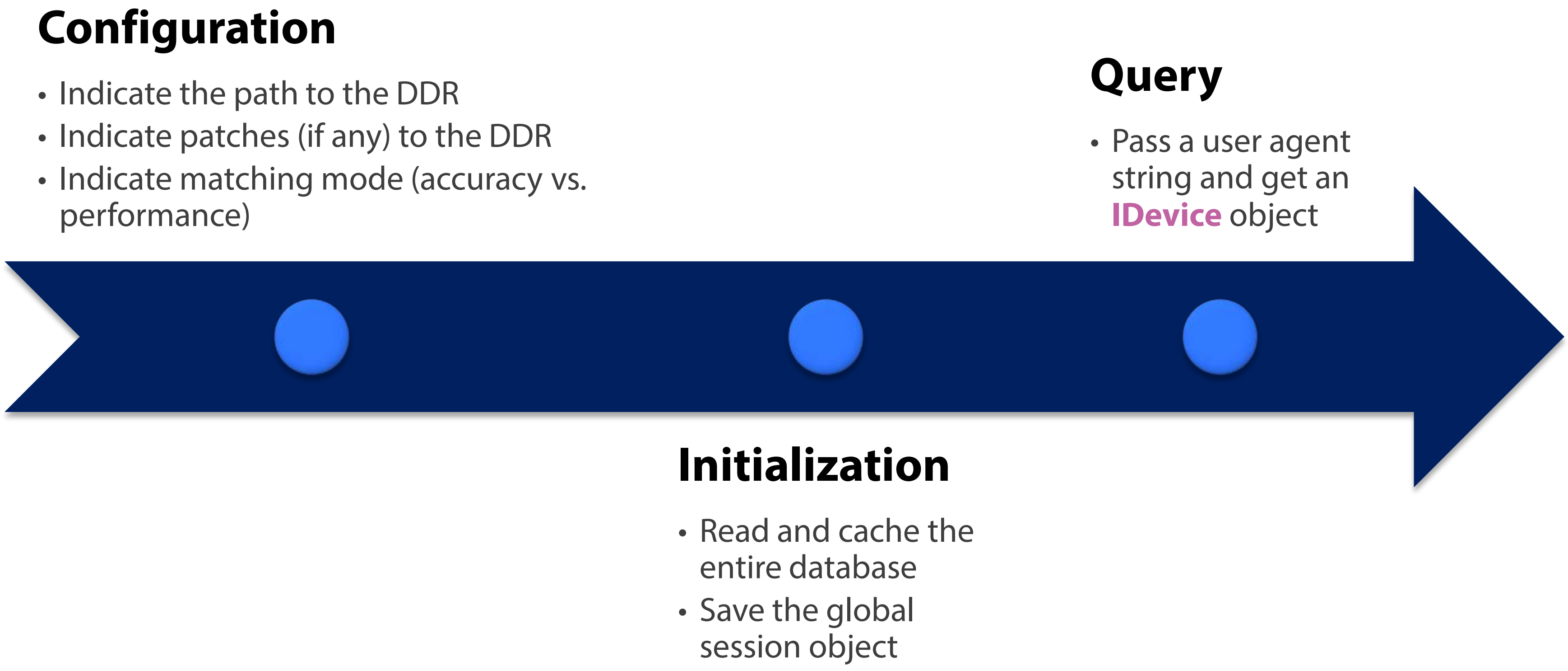
- Up-to-date tables about devices capabilities for mobile analytics

# How Does It Work



# Using WURFL On-premise

Keeping the DDR up-to-date is your responsibility. DDR updated on a weekly basis. Availability and download details depend on license terms.



## Configuration

- Indicate the path to the DDR
- Indicate patches (if any) to the DDR
- Indicate matching mode (accuracy vs. performance)

## Initialization

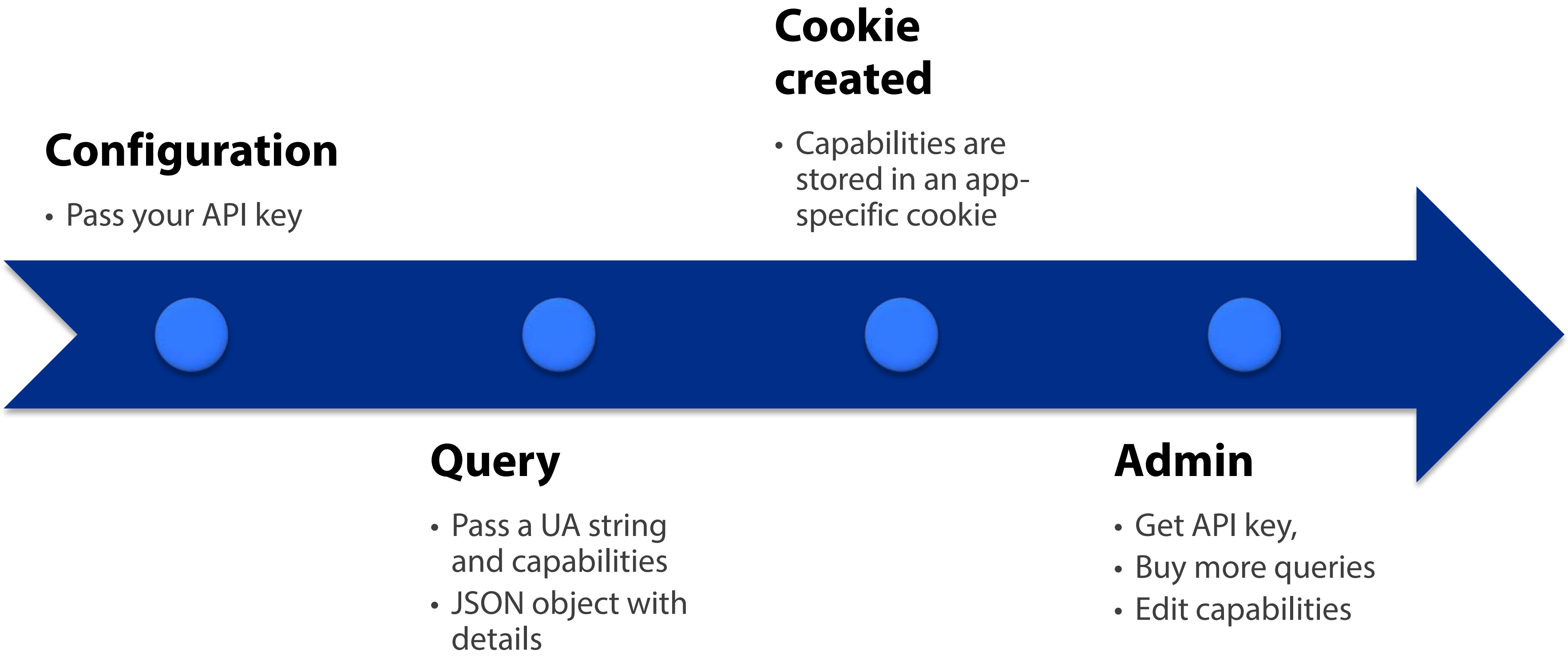
- Read and cache the entire database
- Save the global session object

## Query

- Pass a user agent string and get an **IDevice** object

# Using WURFL Cloud

The DDR is automatically updated on a weekly basis. No effort is required on your end. Higher per-call latency compared to on-premise.



### Configuration

- Pass your API key

### Query

- Pass a UA string and capabilities
- JSON object with details

### Cookie created

- Capabilities are stored in an app-specific cookie

### Admin

- Get API key,
- Buy more queries
- Edit capabilities

# The WURFL Database

**Get the public database from**
*http://wurfl.sourceforge.net*

- **Free** for open-source projects and testing
- **Partially-free** cloud version (limited to just a few capabilities)

**Not allowed to edit/copy the database in any way**

- **Can't access** it using other than one of the WURFL official APIs

# The WURFL Database

- **Flat list of <device> elements**
  - Attribute id uniquely identifies a device (user agent)
  - A commercial device changes UA whenever it gets updated
- **Each device in the database falls back to a parent device**
  - Each <device> element only contains the delta with parent element
  - Multiple root devices: browsers, phone, tablets, smart TVs, robots

```
<devices>
   <device id="..." user_agent="..." fall_back="...">
      <group id="...">
          <capability name="..." value="..." />
      </group>
   </device>
</devices>
```

# Capability Groups

- **Capability is a name/value pair**
  - Value is always a string: “true”, “1”
  - Sometimes pseudo enum types are used: “stylus”, “touchscreen”, …
- **Groups exist only for convenience**

- ajax
- bearer
- chips
- display
- html_ui
- image_format
- pdf
- playback
- product_info
- rss
- security
- sound_format
- smarttv
- sms
- storage
- streaming

```
<group id="css">
   <capability
       name="css_spriting"
       value="true" />
   <capability
       name="css_rounded_corners"
       value="webkit" />
</group>
```

# Patch Files

- **Official way to override content of the WURFL DDR**
  - Make changes to some capabilities without editing the original file

- **Use when you have good reasons for willing to give some capabilities a different value**
  - **Ex:** you want to treat as a tablet a given device that DDR currently classifies as a smartphone
  - **Ex**: add new capabilities not covered (for whatever reason) in the DDR

```
<device user_agent="device that WURFL doesn’t consider a tablet"
        fall_back="generic_mobile" id="mytablet">
     <group id="product_info">
          <capability name="is_tablet" value="true" />
     </group>
</device>
```

# Essential WURFL Capabilities

## Identifying the current device

| Capability | Description |
|---|---|
| device_os | *Name of the OS (ex. Android)* |
| device_os_version | *Version of the OS (ex. 2.2)* |
| resolution_width | *Known screen width in pixels* |
| resolution_height | *Known screen height in pixels* |
| can_assign_phone_number | *Can have a phone number (to distinguish devices using a SIM only to browse the web)* |
| pointing_method | *How to select links: joystick, stylus, touchscreen, clickwheel, default (4-way navigation buttons* |
| brand_name, model_name, marketing_name | *Brand=HTC, model=HTC A8181, marketing=HTC Desire* |

# Essential WURFL Capabilities

## Serve browser-specific content

| Capability | Description |
|---|---|
| viewport_supported | *Supports* viewport *meta tag* |
| image_inlining | *Can display inline images within* IMG *tags* |
| full_flash_support | *Supports Flash* |
| cookie_support | *Supports cookies* |
| preferred_markup | *Preferred type of markup whether HTML or XHTML MP (mobile specific)* |
| png, jpg, gif, tiff, greyscale | *Whether images of specified type can be displayed* |

# Virtual Capabilities

Don’t refer to atomic properties; result from the combination of multiple individual capabilities.

| Capability | Description |
|---|---|
| is_android, is_ios, is_windows_phone | *True if it is any version of Android \| iOS \| Windows Phone.* |
| is_app | *Viewed through WebView within an app* |
| is_full_desktop | *True if it is a desktop browser* |
| is_smartphone | *Smartphone: combination of OS, pointing, screen.* |
| is_touchscreen | *Has a touch screen* |
| is_mobile | *True if it is any mobile device* |
| advertised_device_os | *Ex: Mac OS X* |
| advertised_browser | *Ex: Chrome* |

# Accuracy vs. Performance

Refers to the algorithm the WURFL engine uses to match a user agent string to a known device.

Accuracy **gives as many details as possible**

- For mobile devices, response is always accurate
- Matching mode doesn’t matter

Performance **omits details if it’s a desktop**

- Use if you want to **quickly filter out** desktop browsers

**Set matching mode in configuration**

- **Performance** is default matching mode

# Adding WURFL to an Existing Site

You have an existing site that doesn’t render well on devices.

- **Quickest is writing a separate m-site with**
  - Fewer pages
  - Simpler markup
  - Fewer functions
- **How to bind it to the URL?**
  - Can use a *m.yoursite.com* URL or *www.yoursite.com/mobile*
  - Same full URL wrapped by a HTTP module that redirects to the m-site
  - Need WURFL (or similar) to detect device
  - Need cookies to track choices when you follow URLs in order to bypass the HTTP module

# Summary

- **WURFL: on-premise vs. cloud**
  - DDRs need to be updated frequently
  - Key parameter to consider to make a sound choice
- **Capabilities**
  - Atomic and virtual
  - Hierarchy of values for faster matching
  - Caching is key
- **WURFL de facto standard for DDR**
  - Used by web giants
  - Cross-platform