

The background features a dark grey map of Berlin with a grid overlay. Two large orange arrow shapes point towards each other, framing the central text. Various orange and grey arrow icons are scattered across the map. Labels for 'Prenzlauer Berg', 'Friedrichshain', and 'Rummelsburg' are visible on the map.

# ESRI DEVELOPER SUMMIT

---

10-12 November | Berlin, Germany



# Extending ArcGIS for Server

Jon Satchwell, Esri Switzerland

Cédric Despierre Corporon, Esri France

# Agenda



Introduction



Server Object  
Extensions



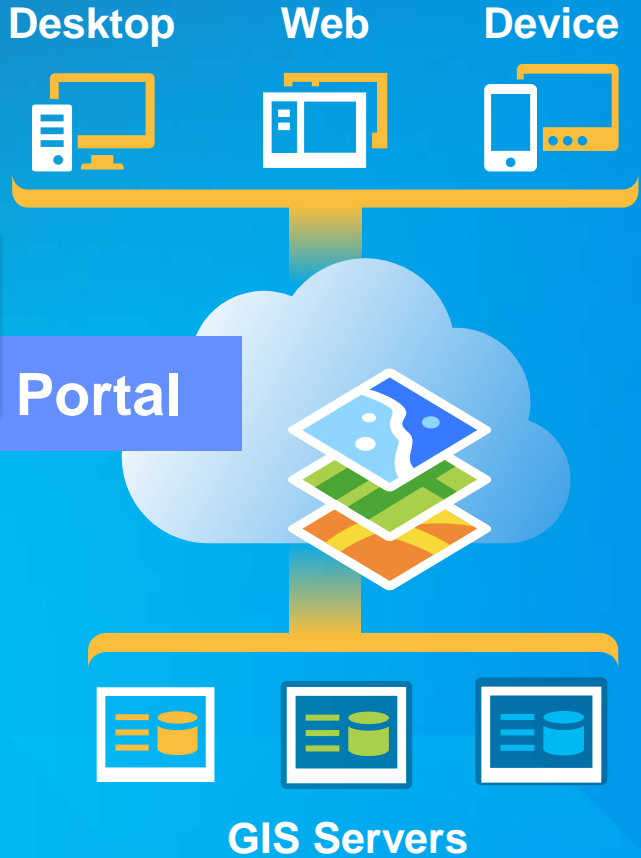
Server Object  
Interceptors



# THE BIG WEBGIS PICTURE

Introduction

**API**



**APPS**

**SHARING**

**SERVICES**

**DATA**

# API



**CREATE**

**CUSTOMIZE**

**EXTEND**

**WEB & MOBILE  
APPLICATIONS**

**TEMPLATES &  
WIDGETS**

Desktop

Web

Device



**CLIENT-SIDE**

Portal



GIS Servers

**SERVER-SIDE**

**CA Fire**  
2014 Fire Map

Find address or place

Home Web AppBuilder for ArcGIS

Theme Map Widget Attribute

- Attribute Table
- Coordinate
- Footer
- Header
- Home Button
- My Location
- Overview Map
- Scalebar
- Search
- Splash
- Swipe
- Time Slider
- Zoom Slider
- Widget
- Widget
- Widget

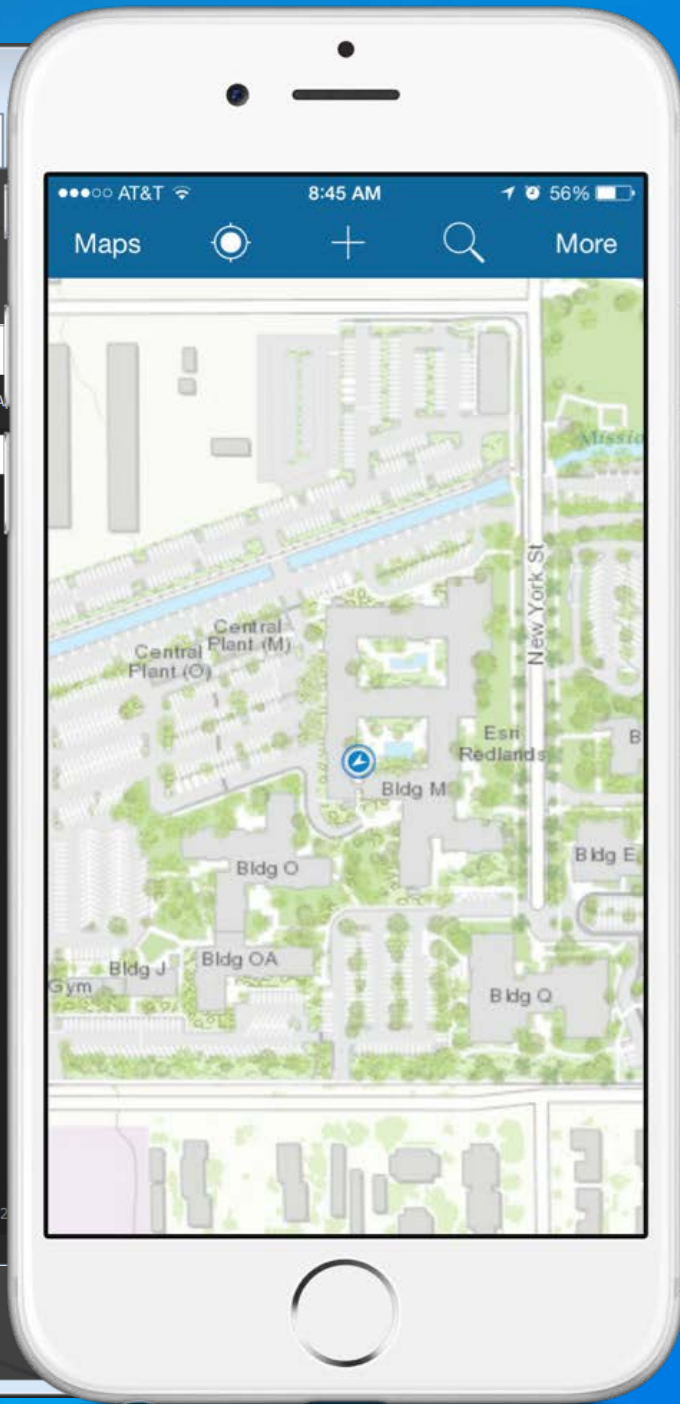
Launch | Previews | Save

**CAFireWebApp** with Web A

Find address or place

1 2 3

100mi 36.378 -12



admin

Salt Lake City

Phoenix

POWERED BY **esri**  
I, HERE, D...

San Diego  
Tijuana



**WEB & MOBILE  
APPLICATIONS**

**TEMPLATES &  
WIDGETS**

Desktop

Web

Device



**CLIENT-SIDE**

Portal



GIS Servers

**SERVER-SIDE**

**CLIENT-SIDE**

Desktop    Web    Device



Portal



**GP TOOLS, SOE & SOI**



GIS Servers

**SERVER-SIDE**

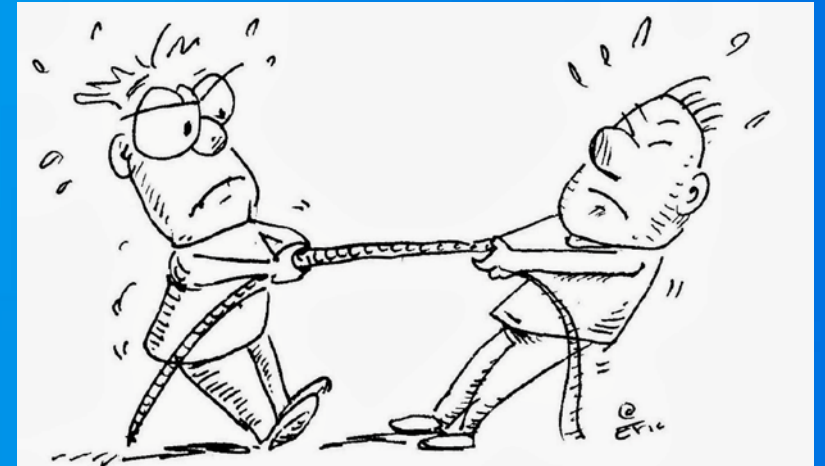
The background features a vibrant blue gradient. On the left side, there are several overlapping geometric shapes in shades of purple and yellow. A small, semi-transparent map fragment is visible within one of the yellow shapes. The main title is centered in white text.

# Extending GIS services

Tools, SOE and SOI

## Why extend GIS services?

- ✓ Extensions run inside the server framework
  - ✓ Security
  - ✓ Scalability
  - ✓ HA
- ✓ Custom business logic
  - ✓ Push round-trip heavy logic to server-side
- ✓ Full access to Esri's GIS component library (arcobjects) to implement core GIS functionality



## Extension types

**ANALYSIS or  
DATA  
MANAGEMENT**

Geo-  
processing  
Tools

Server  
Object  
Interceptors



Server Object  
Extensions

**EXTEND EXISTING  
SERVICES**

**ADD NEW GIS  
CAPABILITIES**

# Out of the box on Server

- Geometry operations example: REST services

ArcGIS REST Services Directory Logged in user : admin | [Logout](#) |

[Home](#) > [services](#) > [Utilities](#) > [Geometry \(GeometryServer\)](#) [Help](#) | [API Reference](#)

[JSON](#) | [SOAP](#)

## Utilities/Geometry (GeometryServer)

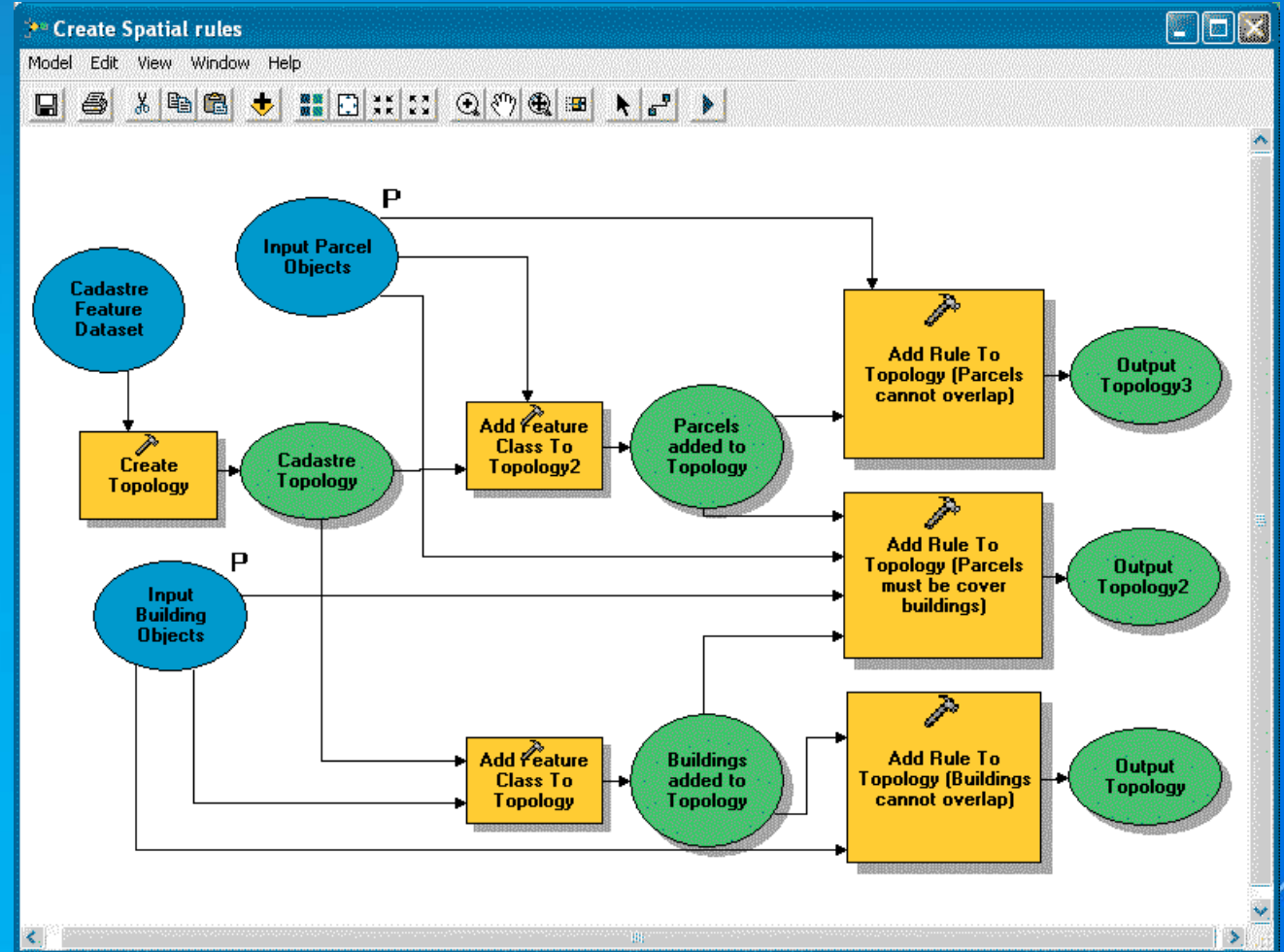
**Service Description:** The Geometry service is used by application developers to perform geometric calculations and web editing

**Supported Operations:** [Areas and Lengths](#) [Auto Complete](#) [Buffer](#) [Convex Hull](#) [Cut](#) [Densify](#) [Difference](#) [Distance](#) [Generalize](#) [Intersect](#) [Label Points](#) [Lengths](#) [Offset](#) [Project](#) [Relation](#) [Reshape](#) [Simplify](#) [Trim Extend](#) [Union](#) [To GeoCoordinate String](#) [From GeoCoordinate String](#) [FindTransformations](#)

**Child Resources:** [Info](#)

# Geoprocessing tools

- ✓ Analysis
- ✓ Data management
- ✓ Back office jobs
- ✓ .NET, Java, Python
- ✓ IDE integration
- ✓ Asynchronous, scalable framework



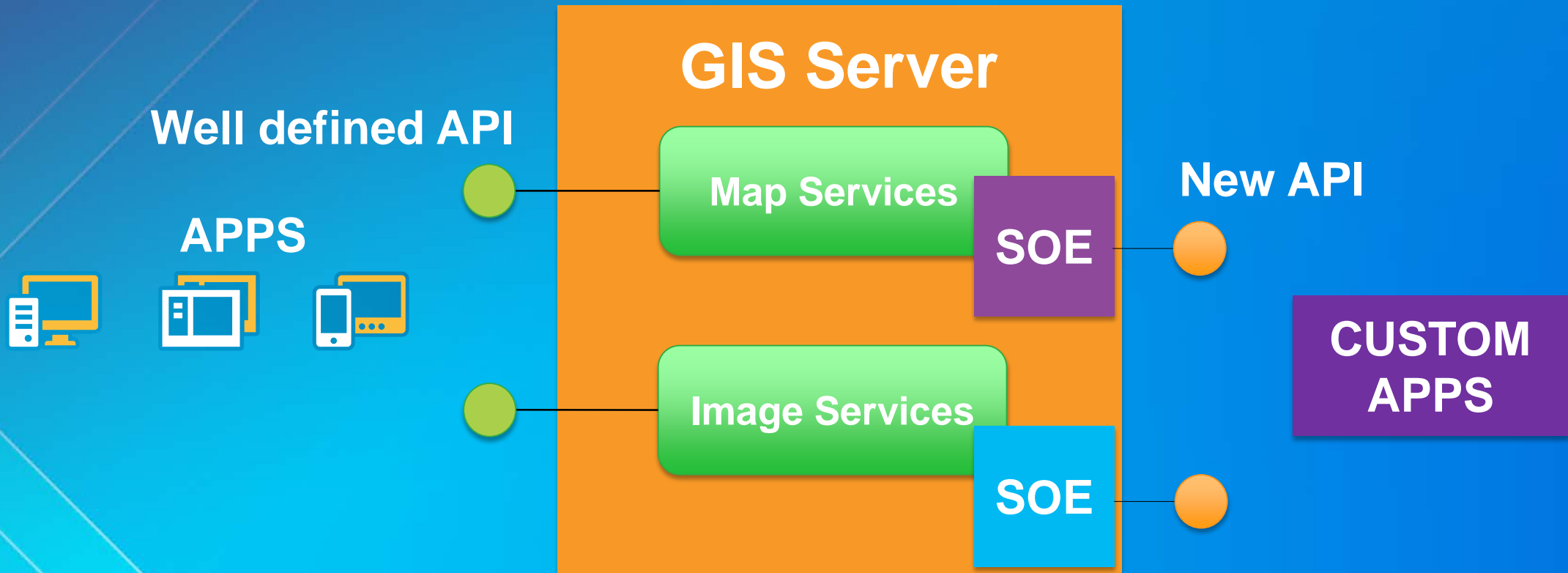


# Server Object Extensions

Adds **NEW** GIS capability to server



# Server object extension (SOE) – internals



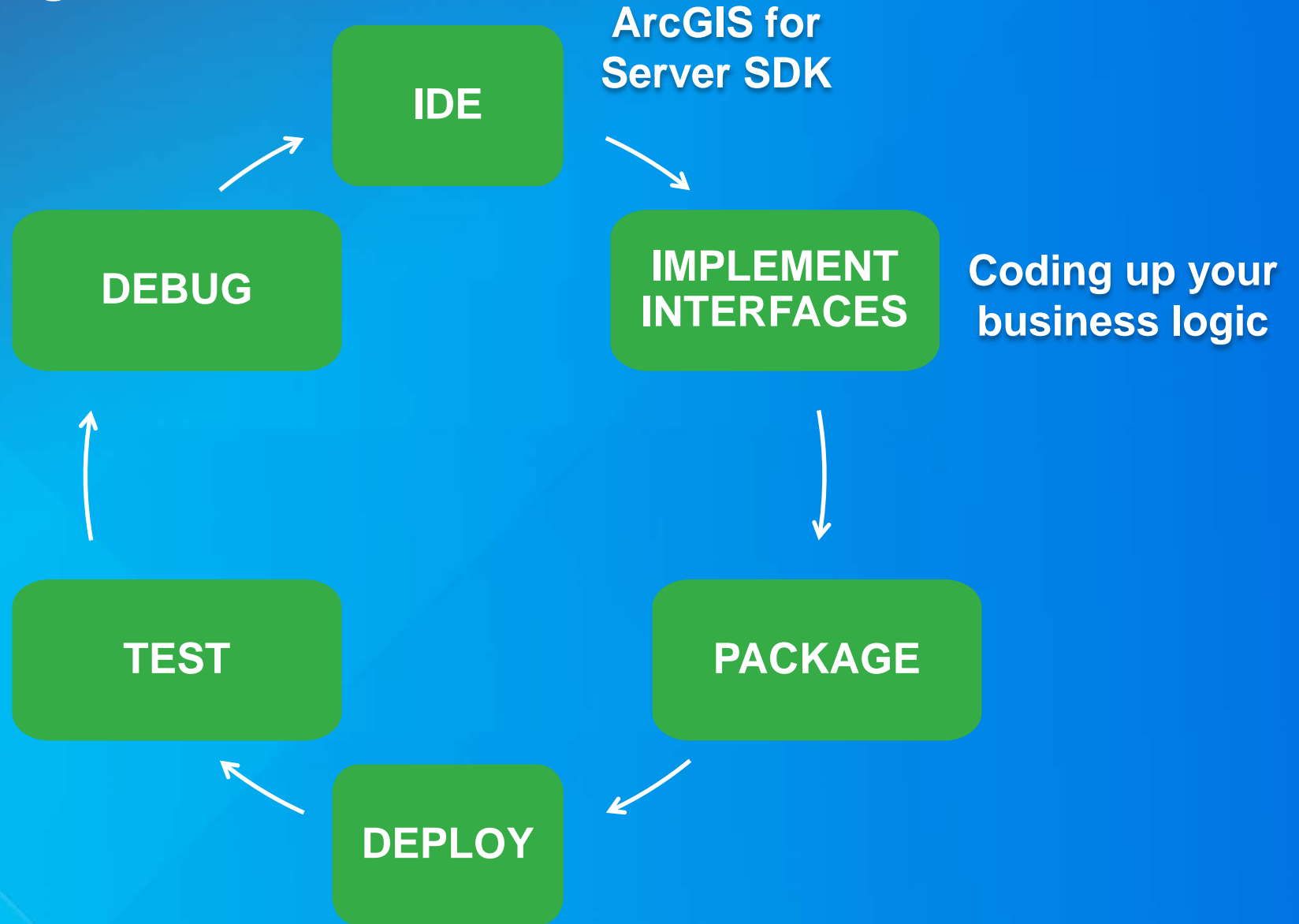
## SOE – use cases

- ✓ **Advanced editing**
- ✓ **Advanced symbology and rendering**
- ✓ **Data extraction**
- ✓ **Advanced raster functions**
- ✓ **Raster management**



*Great way to reduce round-trips in your client apps!*

# Process of extending GIS services



## Developing an SOE

- ✓ **.NET or Java (Java 7)**
  - ✓ **Implement**
    - ✓ **IServerObjectExtension**
    - ✓ **IRESTRequestHandler or**
    - ✓ **IRequestHandler (Soap/binary)**
  - ✓ **Package the implementation (.soe)**
    - ✓ **Zip file**
    - ✓ **Esri-provided tooling generated**
    - ✓ **Contains assemblies or jars**
    - ✓ **Some metadata**
  - ✓ **Deploy to GIS Server through Manager application**
  - ✓ **Enable SOE for a service**
- SDK contains base classes that already implement these interfaces for you!

## SOE – interfaces to implement

**TIP: This should execute very quickly!**

Name	Functions	Purpose
IServerObjectExtension	init(...) shutdown()	Initialization of SOE Cleanup
IRESTRequestHandler	getSchema() handleRESTRequest(...)	Define the API as resources and operations Handle all REST resources and operation requests
IRequestHandler2	handleStringRequest(...) handleBinaryRequest(...)	Handle SOAP requests Handle “binary” requests from ArcMap

**Most server object extensions are REST API based!**

## SOE – tips and best practices...

- ✓ **Always start with the template in your IDE**
  - ✓ Boiler-plate code
- ✓ **Easier to implement REST API**
  - ✓ Services directory view is auto-generated
- ✓ **Custom property configuration pages**
- ✓ **Follow the debugging guide in server help**

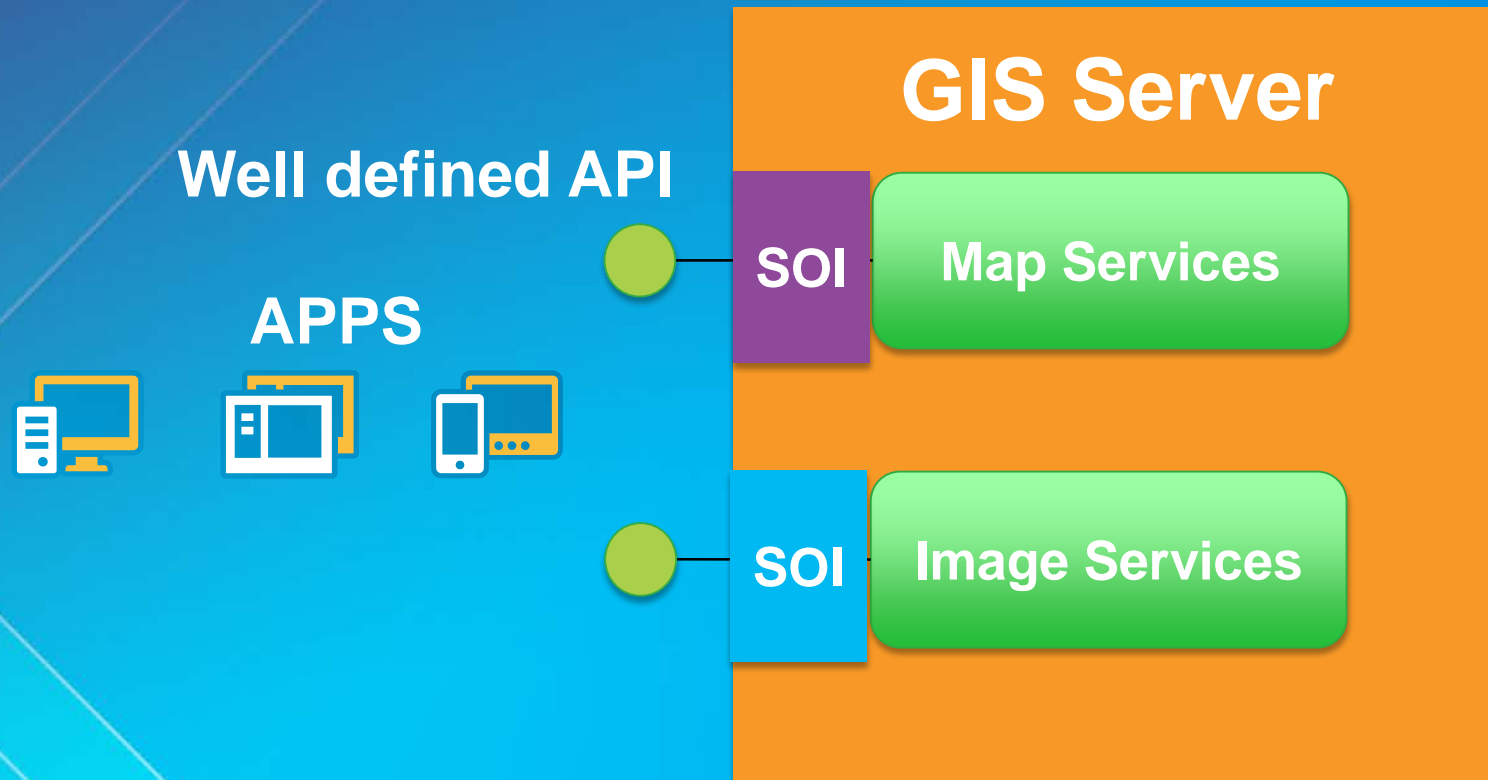


# Server Object Interceptors

New at 10.3.1

Extends **EXISTING** capabilities

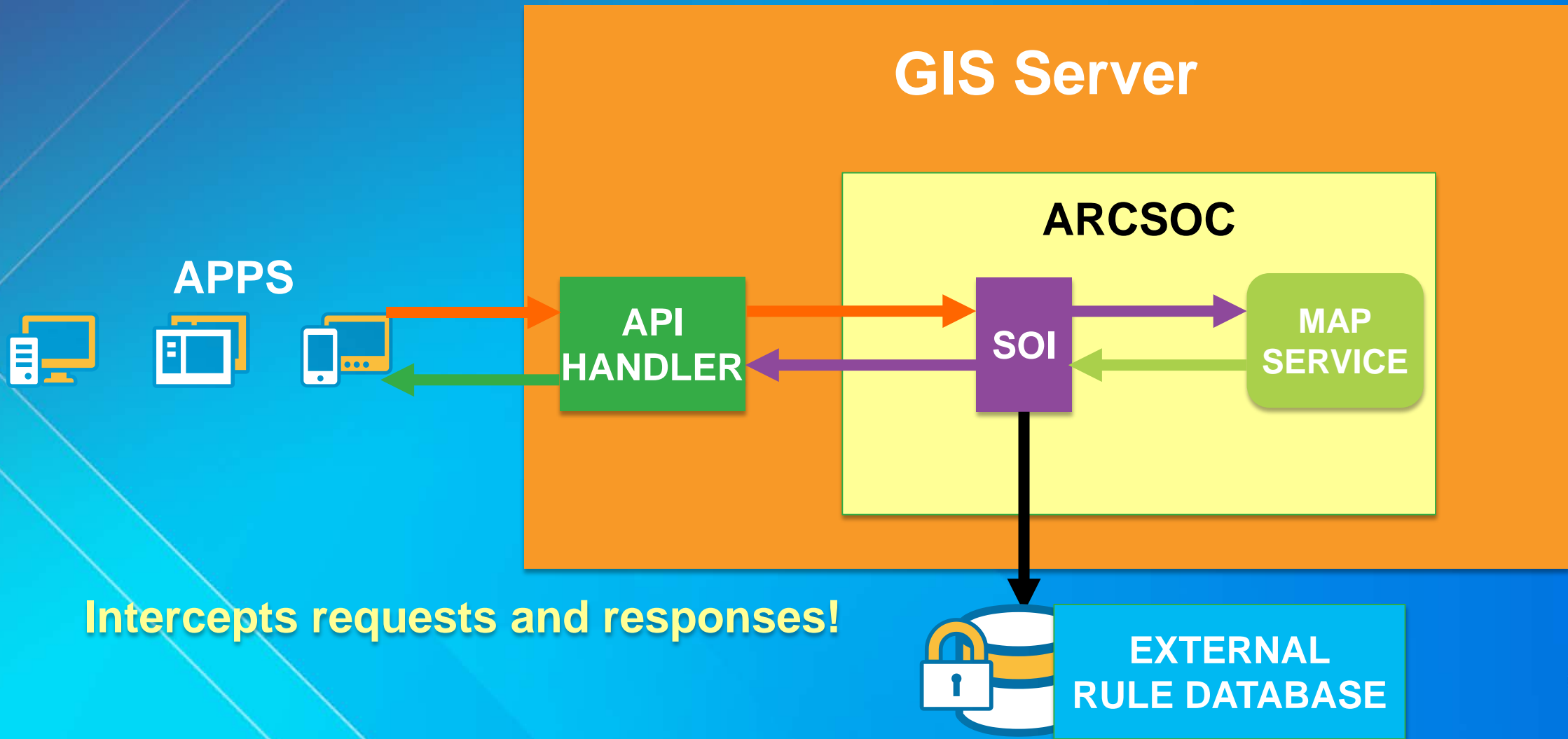
# Server object interceptors (SOI) – internals



*Existing applications continue to work!*



# SOI – execution model



## SOI – use cases

- ✓ **Advanced security**
  - ✓ Call into external systems
- ✓ **Enforce business rules**
- ✓ **Add quality assurance**
  - ✓ Trigger back-office workflows upon receiving updates



*Custom filtering logic for existing apps!*

## Developing an SOI

- ✓ Decide on interceptors for Map or Image services
  - ✓ .NET or Java (Java 7)
  - ✓ Implement
    - ✓ IServerObjectExtension
    - ✓ IRESTRequestHandler
    - ✓ IRequestHandler
    - ✓ IWebRequestHandler
  - ✓ Package the implementation (.soe)
  - ✓ Deploy to GIS Server through Manager application
  - ✓ Enable SOI for a service
- SDK contains base classes that already implement these interfaces for you!

## SOI – interfaces

Name	Functions	Purpose
IServerObjectExtension	init(...) shutdown()	Initialization of SOE Cleanup
IRESTRequestHandler	getSchema()  handleRESTRequest(...)	Define the API as resources and operations Handle all REST resources and operation requests
IRequestHandler2	handleStringRequest(...) handleBinaryRequest(...)	Handle SOAP requests Handle “binary” requests from ArcMap
IWebRequestHandler	handleStringWebRequest(...)	Handle OGC requests

**Interfaces for extensions and interceptors are the same but behavior/logic would have to be different!**

## SOI – tips and best practices...

- ✓ **Always start with the template in your IDE**
  - ✓ Boiler-plate code
  - ✓ Default delegation based behavior
- ✓ **Utility functions**
- ✓ **API cannot be changed**
- ✓ **For security interceptors:**
  - ✓ Block all operations first
  - ✓ Iteratively allow operations and resources
- ✓ **Does not intercept tiled map services and hosted feature services**



# DEMO

Building some SOI use-cases

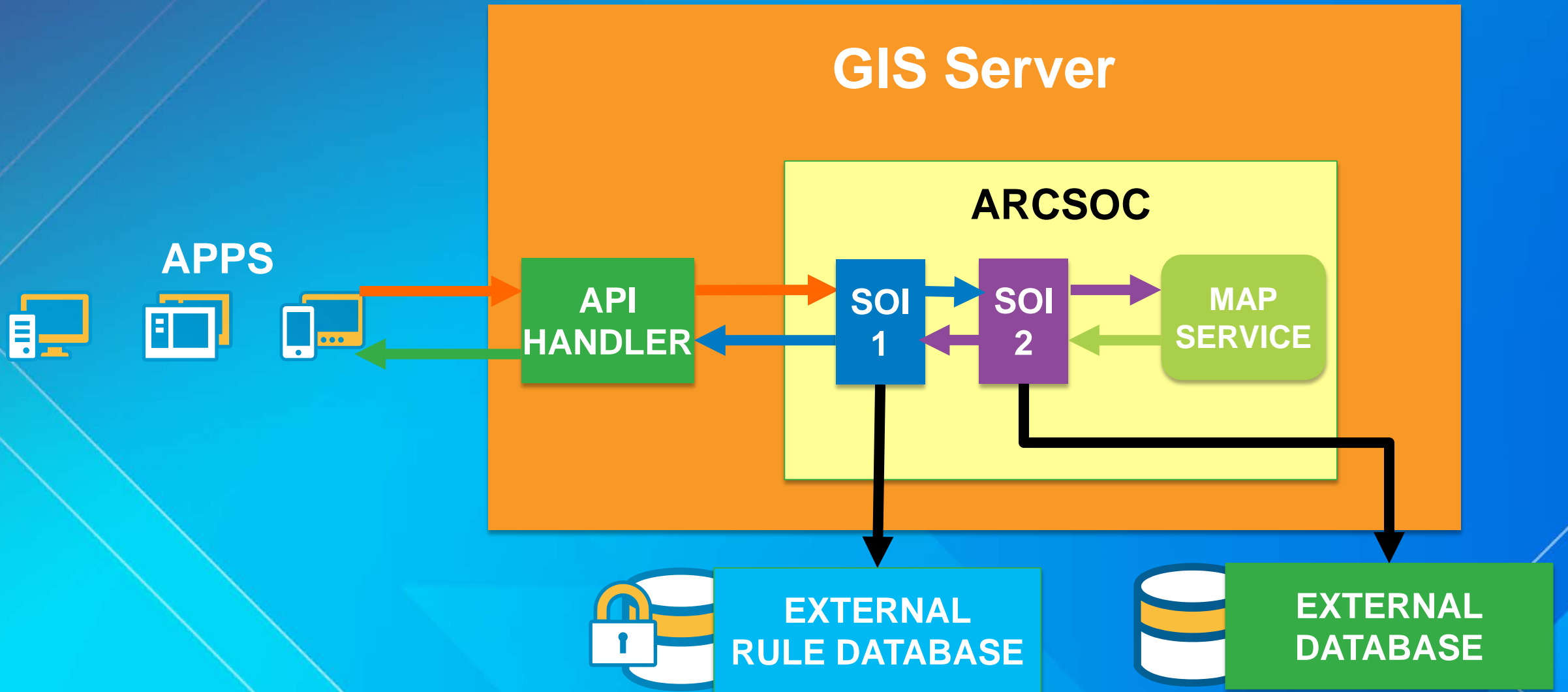
Code is here : <https://github.com/ceddc/devsummit-berlin-2015-SOI>

## SOI chaining (in the next release)

- ✓ Chain multiple SOIs together
- ✓ Allows for in-house and third party interceptors
- ✓ SOIs can be repeated
  - ✓ Particularly useful for security specific SOIs



# SOI chaining – execution model





## In summary...

- ✓ **Server Object Interceptors**
  - ✓ New at 10.3.1
  - ✓ Change the behavior of existing GIS services
- ✓ **Server Object Extensions**
  - ✓ Add new GIS functionality (API)

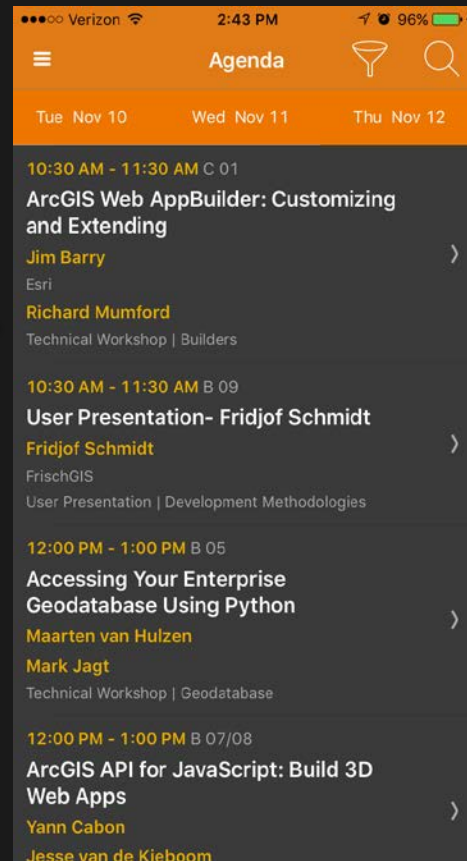


# Please Take Our Survey!

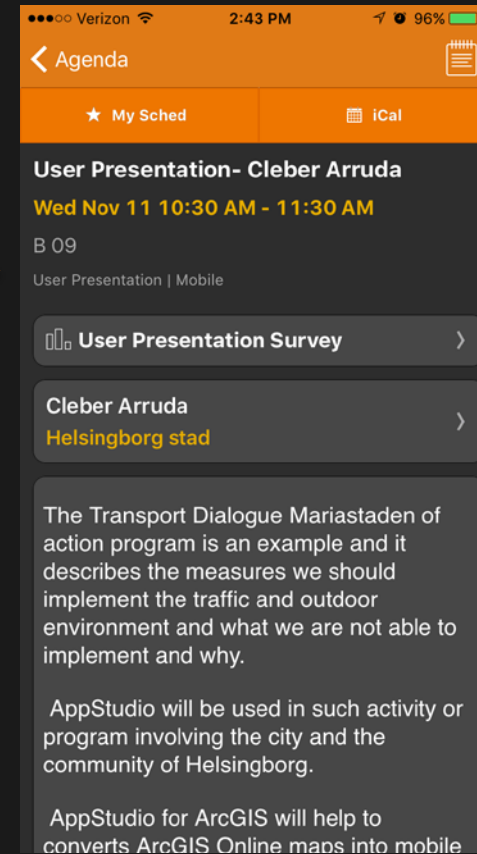
Download the Esri Events app  
and find your event



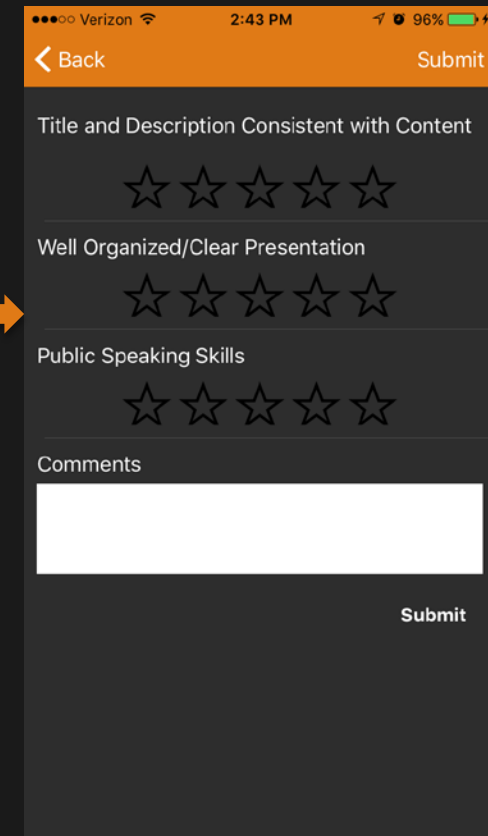
Select the session you  
attended



Select  
"User Presentation Survey"  
or  
"Technical Workshop Survey"



Complete Answers  
and Select "Submit"





esri®

BERLIN

Prenzlauer Berg

Friedrichshagen

Friedrichshagen

Friedrichshagen