Using

the ArcGIS JS API

PostGIS,

SDE

(& ArcGIS Server)

the "Wrong" Way

Esri Dev Summit, Wed., Mar. 9, 2016, 3:05 PM – 3:30 PM, Mesquite B - Nick Floersch & Alan Hammersmith

Two Amazing and Life-changing Stories...

Story 1: A Few Years Ago...

Uhhhh...

Nick and Alan

Nick and Alan

Story 2: Meet The Teams @ Dev Summit 2012

The Project

Actually Suggested Application Title

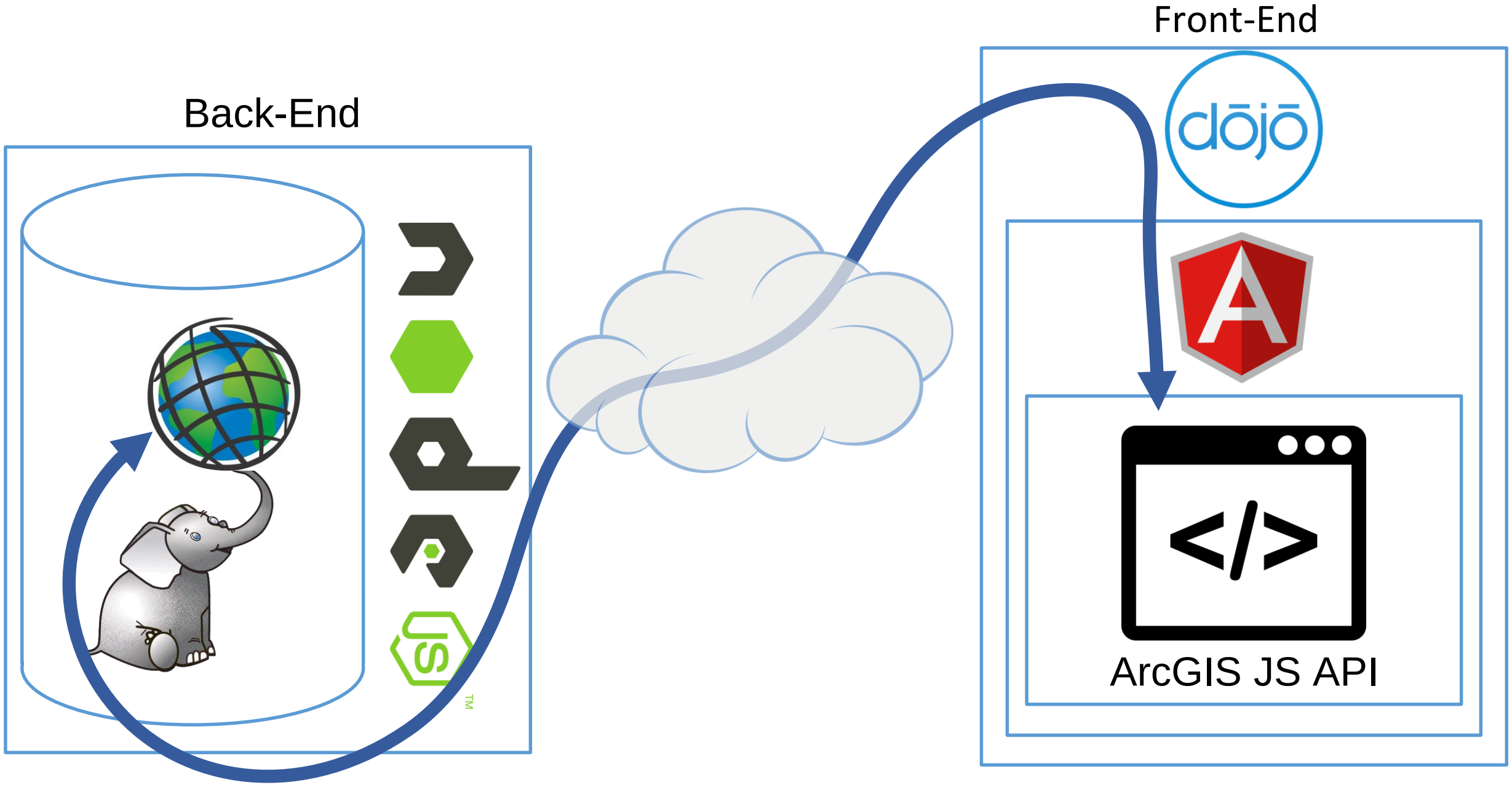
"Multi-Partner Agricultural Conservation Practice Tracking and Planning Geospatial Database"

The Requirements That Drove Our Solution

Flickr: Infrogmation

Why Not Use **Esri Best Practices**?

The ‘Solution’ At 1:10,000,000 scale

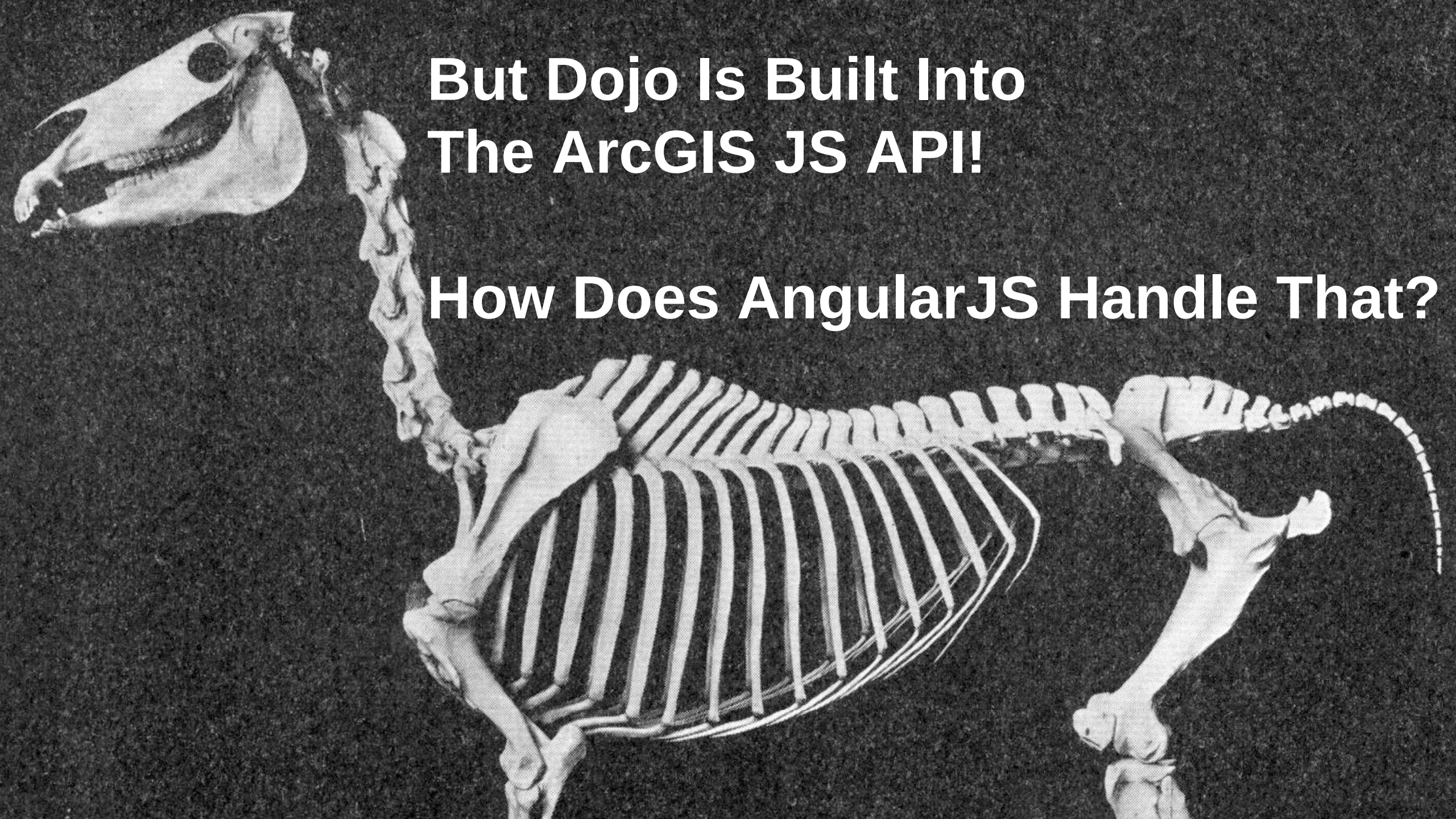


The Front End

The ***Esri*** Way...

Probably
ArcGIS JS API
&
Dojo
(& maybe some jQuery)

Front-End .. If Not The Esri Way, What Way?

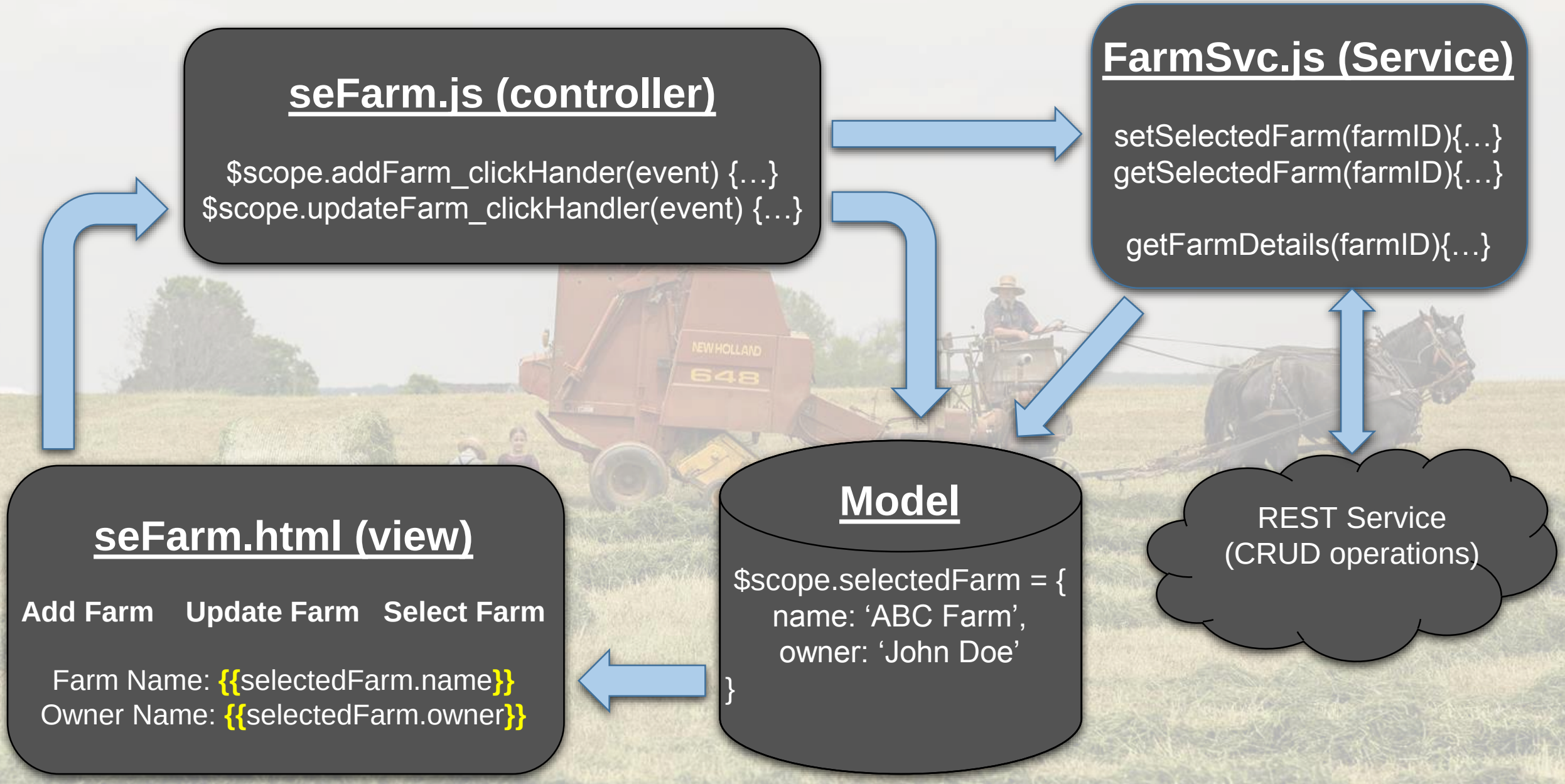


But Dojo Is Built Into The ArcGIS JS API!

How Does AngularJS Handle That?

Pros/Cons of the AngularJS Choice

Our Farm **Front-End** (*only!*) Code-Structure In AngularJS



The Back-End

What is the
Esri Way to do
a Back-End?

Flickr: paul_everett82

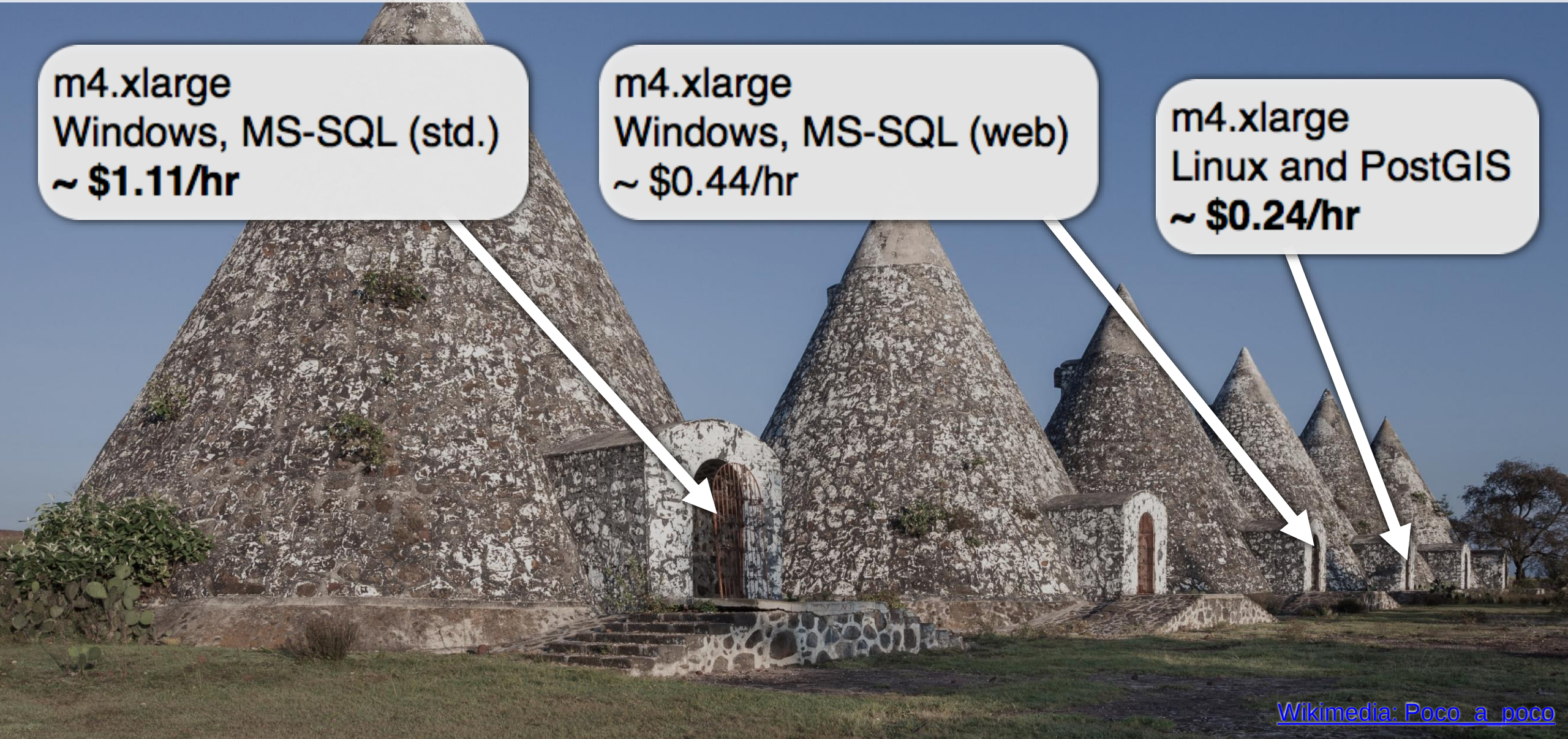
The Super Nifty Wrong Way to “Back-End”

Part 1: Data Storage

Wikimedia: Huwmanbeing

The Super Nifty Wrong Way to “Back-End”

Part 1a: *Cheap* Data Storage



The Super Nifty Wrong Way to “Back-End”

Part 2: Database Spatial Processing Functionality

Flickr: grggssmr

The Super Nifty Wrong Way to “Back-End”

Part 3: Data Storage Format Is Important

Flickr: pagedooley

Flickr: saida_haechler

The Super Nifty Wrong Way to “Back-End”

Part 4: Unload Logic From Front-End

```
CREATE FUNCTION ag.getfarm(token uuid, fid integer)
RETURNS TABLE(
farmid integer, farmname text, county text, personid integer, lastname text) AS
  $BODY$
  BEGIN
    RETURN QUERY
    SELECT distinct farm.farmid, farm.farmname, farm.county, [...]
    FROM ag.farm
    LEFT JOIN (Select persontype.relatedtoid, person.personid, person.lastname, [...]
      FROM ag.persontype, ag.person
      WHERE persontype.personid = person.personid
        AND persontype.persontype = 133 -- 'farm owner'
        AND persontype.relatedto = 'farm') p
      ON farm.farmid = p.relatedtoid
    WHERE farm.farmid = $1
  END;
  $BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100
  ROWS 1000;
```

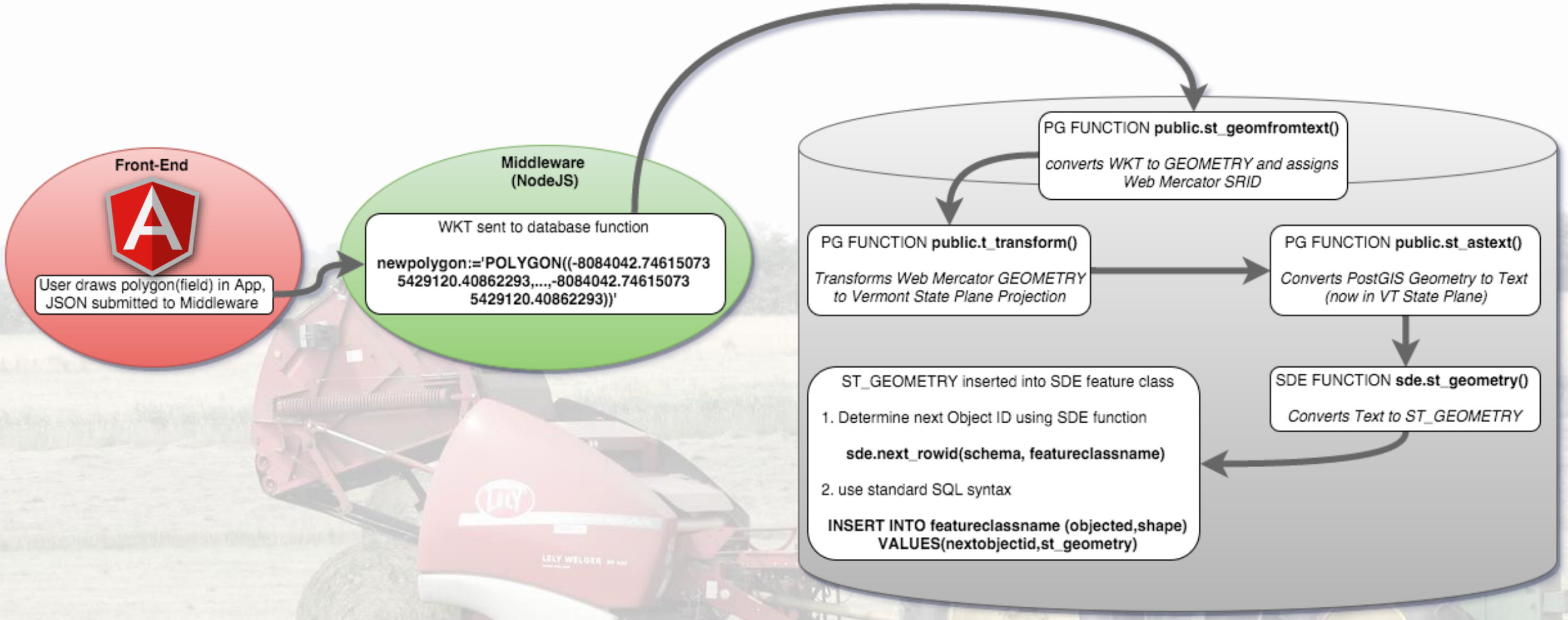
Wikimedia: Användare:PER9000

The Super Nifty Wrong Way to “Back-End”

Part 5: Data Transformation

The Super Nifty Wrong Way to “Back-End”

Part 5: Data Transformation

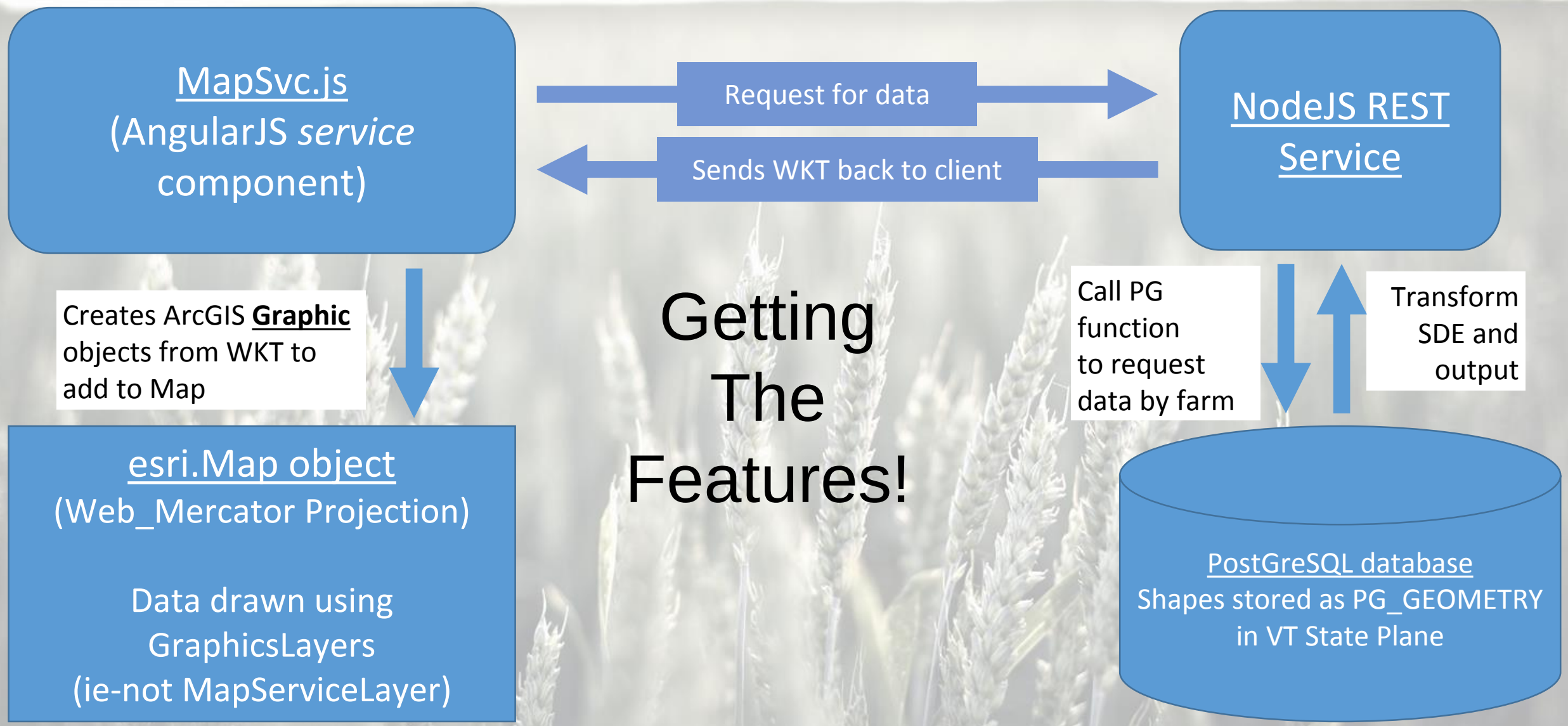


The Super Nifty Wrong Way to “Back-End”

Part 6: The Front-End To The Back-End

Flickr: queenslandstatearchives

The Super Nifty Wrong Way to “Back-End”
Part 6a: Front Requests Features



The Super Nifty Wrong Way to “Back-End”

Part 6b: Convert WKT to Esri Graphic

PG database object representation as JSON, using WKT:

```
{
    farmname: ‘ABC Farm’, owner: ‘John Doe’, geometry: ‘POINT (123.456, 987.654)’
}
```

```
var json = {
    attributes: {
        farmname: 'ABC Farm',
        owner: 'John Doe',
        geometry: 'POINT (123.456, 987.654)'
    },
    geometry: {
      x: 123.456,
      y: 987.654,
      spatialReference: {wkid: 102100}
    },
    symbol: getSymbol(type)
};
var g = new esri.Graphic(json);
map.graphics.add(g);
```

The Super Nifty Wrong Way to “Back-End”

Part 6c: Convert WKT to Esri Graphic

<u>POINT</u>

“POINT (30, 10)”

{ x: 30, y: 10, spatialReference: {wkid: 102100} }

```
// geomStr is "(30 10)"
geomStr = geomStr.replace(/\(/g, '').replace(/\)/g, '');// "30 10"
var pointArr = geomStr.split(' ');                      // [30,10]
geomObj = {
     x: parseFloat(pointArr[0]),
     y: parseFloat(pointArr[1]),
     spatialReference: { wkid: 102100 }
};
g.geometry = geomObj;
return new Graphic(g);
```

The Super Nifty Wrong Way to “Back-End”

Part 6d: Convert WKT to Esri Graphic

LINE

WKT: “LINESTRING (30 10,10 30,40 40)”

ArcGIS: { paths: [[[30,10], [10,30], [40,40]]], spatialReference: {wkid:102100} }

```
// geomStr is "(30 10,10 30,40 40)"
geomStr = geomStr.replace(/\(/g, '[').replace(/\)/g, ']');
                                          // [30 10,10 30,40 40]
geomStr = geomStr.replace(/,/g, '],['); // [30 10],[10 30],[40 40]
geomStr = geomStr.replace(/ /g, ',');   // [30,10],[10,30],[40,40]
geomStr = '[[' + geomStr + ']]';        // [[[30 10],[10 30],[40 40]]]
geomObj = JSON.parse(geomStr);
g.geometry = { paths: geomObj, spatialReference: { wkid: 102100 } };
return new Graphic(g);
```

The Super Nifty Wrong Way to “Back-End”
Part 6e: Convert WKT to Esri Graphic

POLYGON

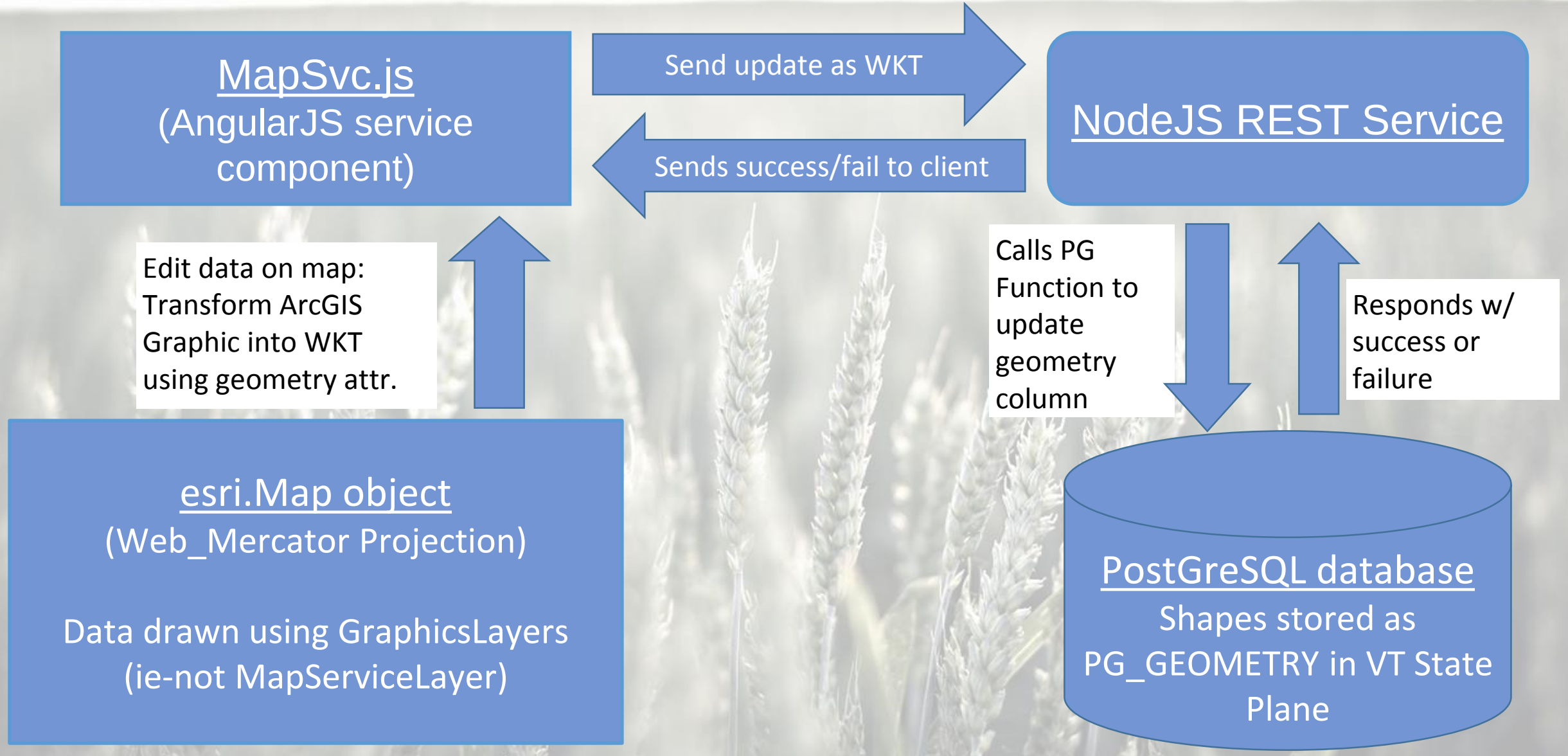
WKT: ”POLYGON ((35 10,45 45,15 40), (20 30,35 35,30 20,20 30))”

ArcGIS: { rings: [[[35,10],[45,45],[15,40]],[[20,30],[35,35],[30,20],[20,30]]], spatialReference: {wkid: 102100} }

```
// geomStr is "( (35 10,45 45,15 40), (20 30,35 35,30 20,20 30) )"
geomStr = geomStr.replace(/\( /g, '[');
geomStr = geomStr.replace(/\(/g, '[').replace(/\)/g, ']');
geomStr = geomStr.replace(/,/g, '],[');
geomStr = geomStr.replace(/ /g, ',');
geomStr = '[' + geomStr + ']';
geomObj = JSON.parse(geomStr);
g.geometry = { rings: geomObj, spatialReference: { wkid: 102100 } };
return new Graphic(g);
```

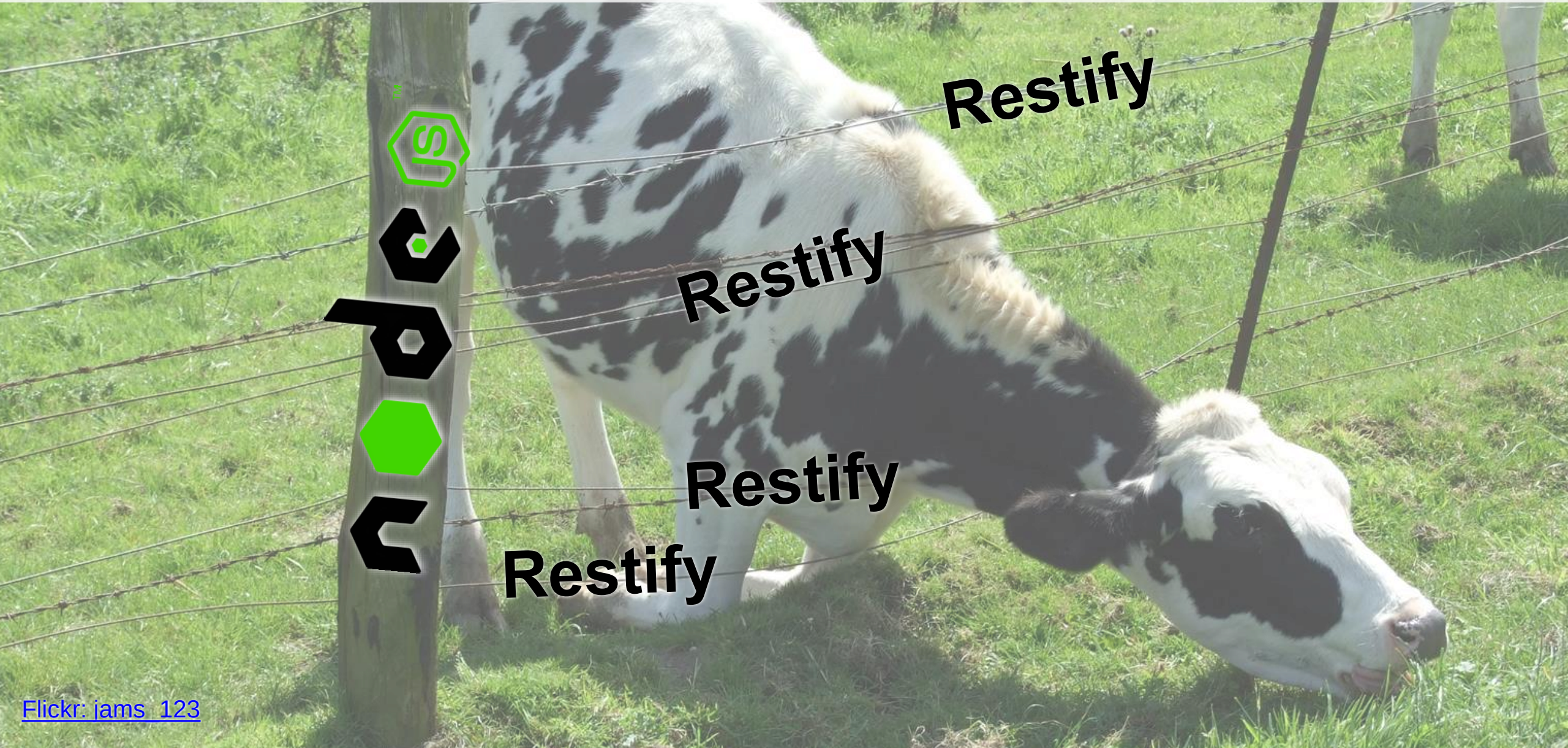
The Super Nifty Wrong Way to “Back-End”

Part 6e: Sending Graphics To Back-end



The Super Nifty Wrong Way to “Back-End”

Part 7: the Middleman with NodeJS, not ArcGIS Server



Flickr: jams_123

The Super Nifty Wrong Way to “Back-End”

Node at REST

https://myserver/**getFarm/4312** (+ JSON ‘params’ object)

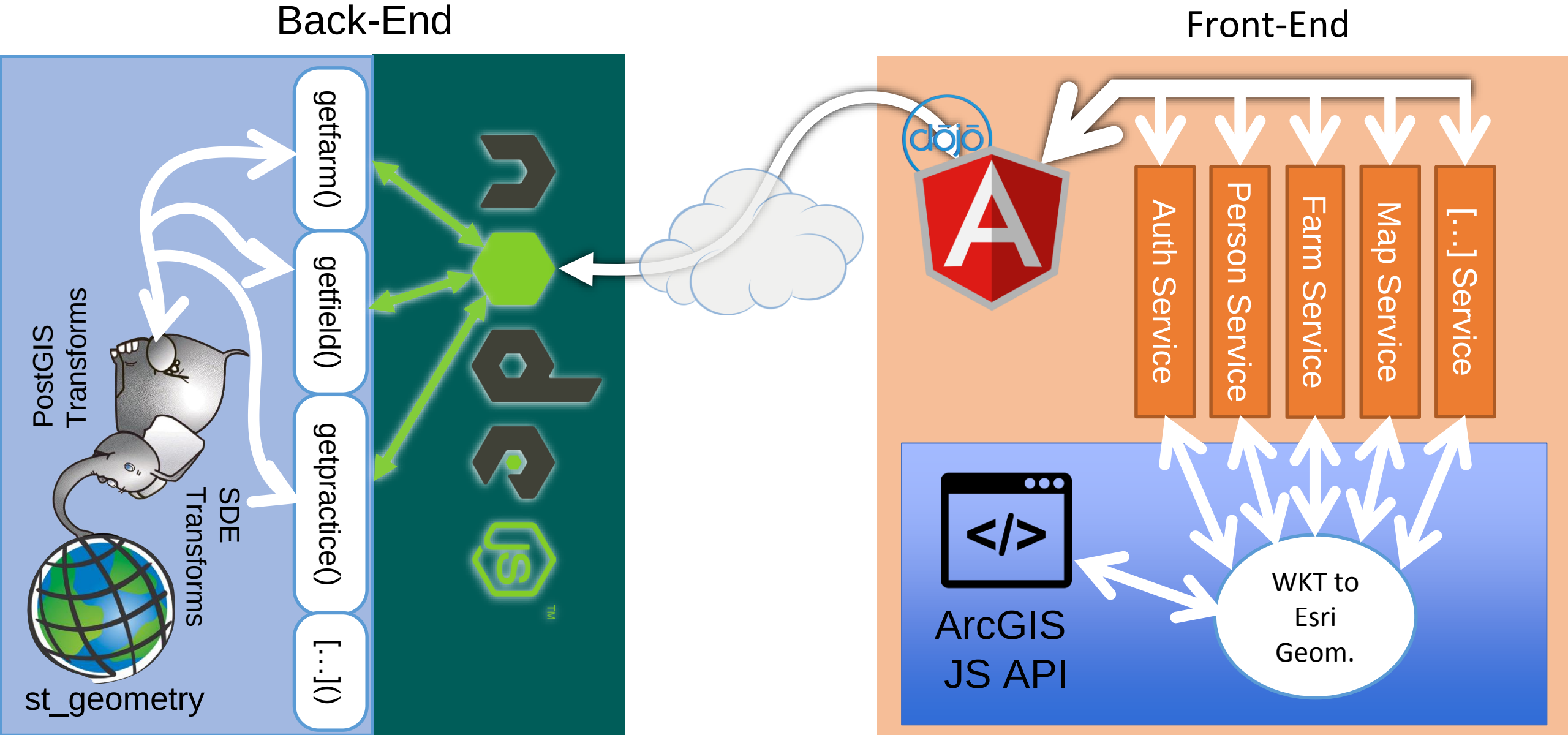
```
app.post('/getFarm/:farmId',
  authorize,        // Validate Token
  setupCORSCrap,    // Setup CORS security rules
  function(req, res, next) {
    var authorization = req.headers.authorization;      // Extract Auth Token
    tToken = authorization.split(" “)[1];               // Extract Auth Token
    tToken = new Buffer(tToken, 'base64').toString("ascii"); // Decode Token

    if (req.params == undefined) req.params = {}; // Make sure to pass in
                                                  // empty params at least
    req.params['token'] = tToken;  // Insert Token To Request Params
    db.getFarm(req.params.farmId, req.params, res, next); // Call func to tell
                                                          // DB what to do
  }
);
```

The Super Nifty Wrong Way to “Back-End”
a RESTful power-nap

```
// INSIDE the db.getFarm() function…
var queryConfig = { // Query Configuration Object
    text: "SELECT * FROM ag.getfarm(token:=$1,fid:=$2)",
    values: [ params.token, params.farmId ]
    };
PGclient.query(  // Using PG module to talk to PostGreSQL
    queryConfig,
    function (err, result) {  // Callback When Query Done
        if (err) {
            done();
            return 'could not connect to postgres';
        }
        var queryResult = result.rows;  // Actual Results
        done();
        res.json(200,queryResult);  // Sends Results To Front-End As JSON
        return next();
    }
);
```

The Super Nifty Wrong Way to “Back-End”

Roundup… what did we just show you?



SO Why Use Esri?

Remember The Title of This Presentation States We ***Are*** Using “ArcGIS Server”… ?


SO Why Use Esri?
What do we lose in this whole approach?

SO Why Use Esri? …

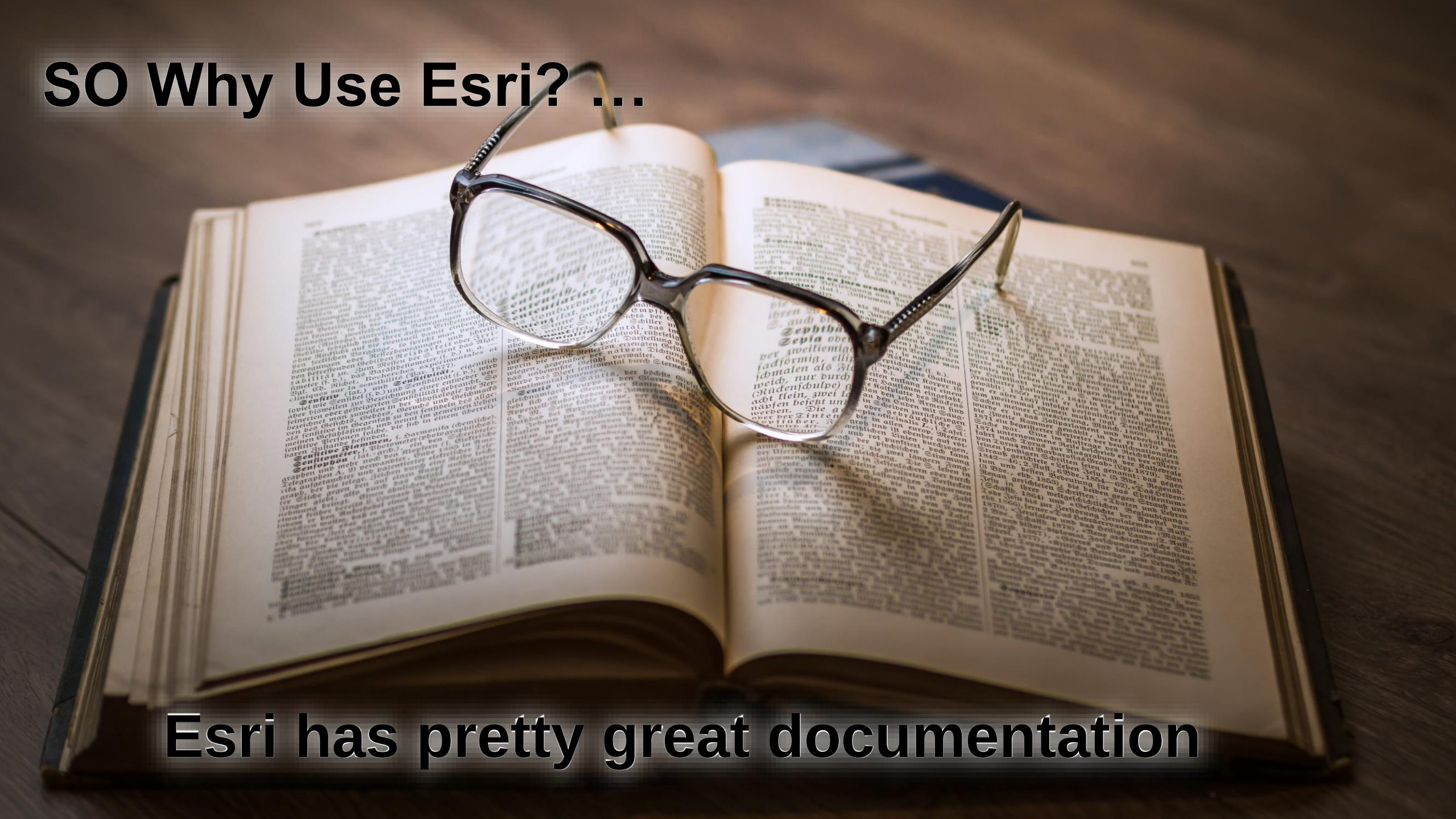
Web GIS With Esri's Tools

Wikimedia: Royalbroil

Our Tool

Wikimedia: Chmee2

SO Why Use Esri? …



Esri has pretty great documentation

Nick Floersch
Alan Hammersmith

Stone Environmental, Inc.
Montpelier, Vermont, USA

tech@stone-env.com

@StoneEnv_AIM

www.stone-env.com/aim

https://github.com/HackVT-Stone/devsummit