A Cotton Irrigator's Decision Support System Using National, Regional and Local Data

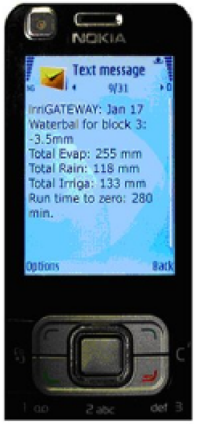
ISESS 2015, Melbourne

Jamie Vleeshouwer, Nicholas J. Car, John Hornbuckle

26 March 2015

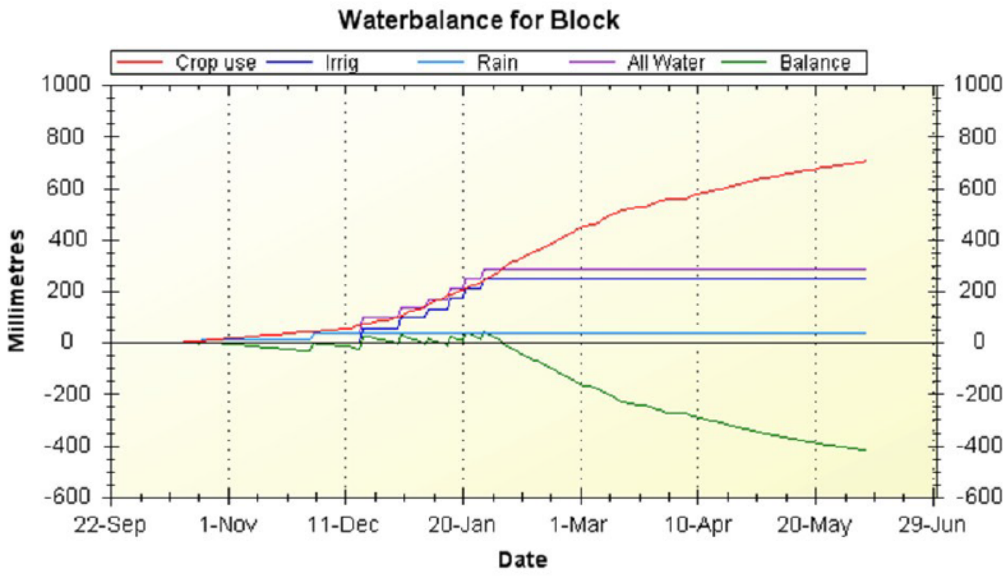
Irrigation in the cotton industry

- Background
  - Water is scarce
  - Irrigation entitlements for regions
  - Farmers maximise revenue by getting best return on water “more crop per drop”
- IrriSATSMS
  - Weather/Sattelite based irrigation decision support system
  - Provide information on how much to irrigate
  - SMS User Interface
  - Has become outdated
- IrriSAT – Next Gen
  - Use modern technology / automation
  - Add forecasting (7 days) / benchmarking functionality



How much water should I use?

- Water balance
  - Rain + Irrig – CropWaterUse = 0
  - Too much = inefficient
  - Too little = restricted growth



Actual Crop Water Use

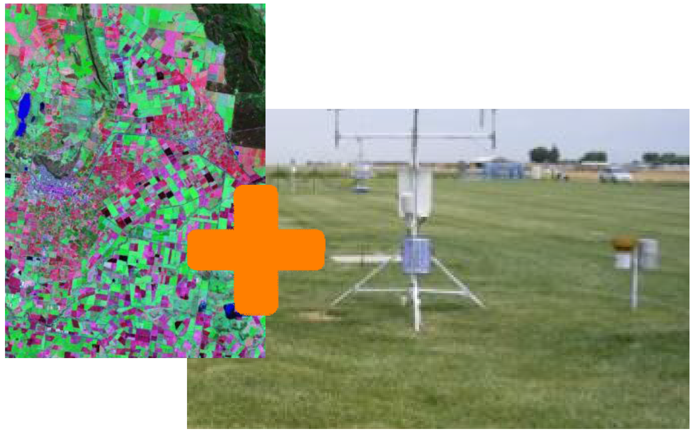
Reference Water Use (Weather Station / SILO)

Crop Coefficient (Relates your crop to the reference crop)

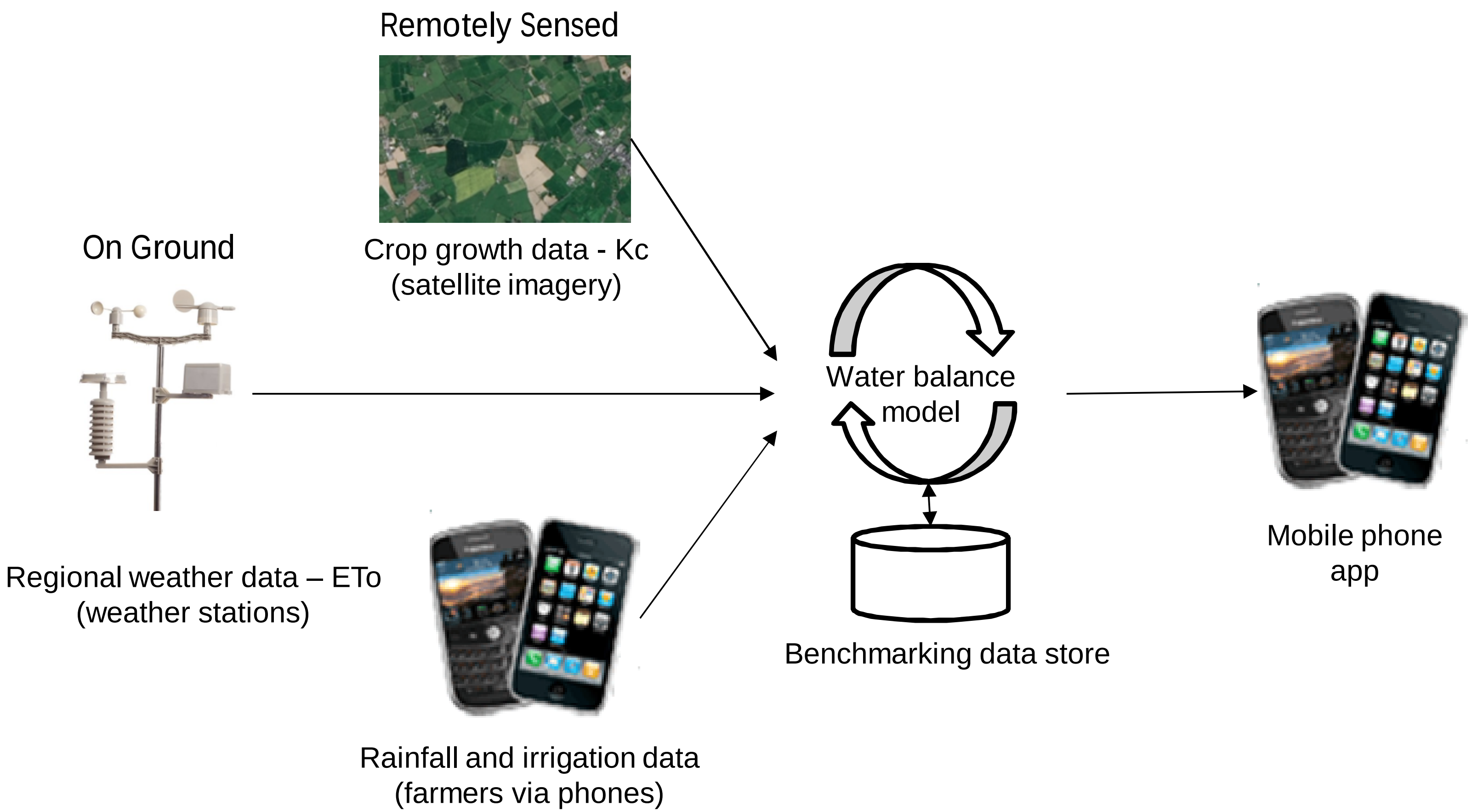
$$ETc = ETo \times Kc$$

What are we doing about it?

- IrriSAT - weather-based irrigation scheduling service
- Satellite imagery determines crop coefficients (Kc) to calculate crop water requirements and provides customised scheduling information
- Irrigation scheduling with:
  - low-cost
  - wide coverage
- Add value - not replace existing methods
- Daily water balance updated in real time
- Targeting smartphone interfaces as well as web



How will it work?



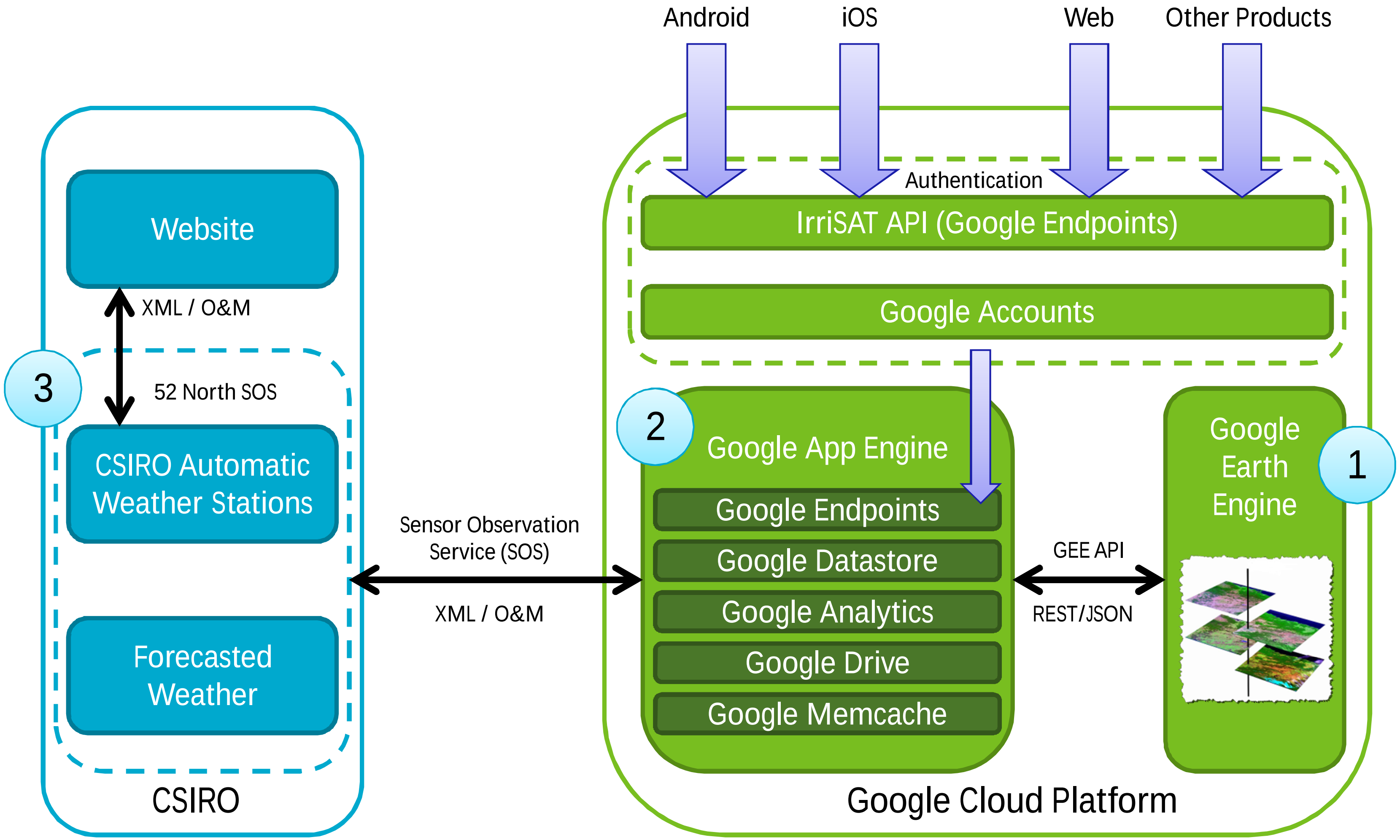
What could we improve upon?

- What didn't work well in the previous system
  - Satellite processing was all manual (lots of time, effort, data storage)
  - Data entry errors (SMS autocorrect interfering)
  - Had commercial interest but too complex to transfer
  - Tightly coupled

- How technology has evolved
  - Cloud computing
  - Web services
  - High performance computing
  - Increased usage of smartphones / tablets

After many thoughts, discussions and experimentation...

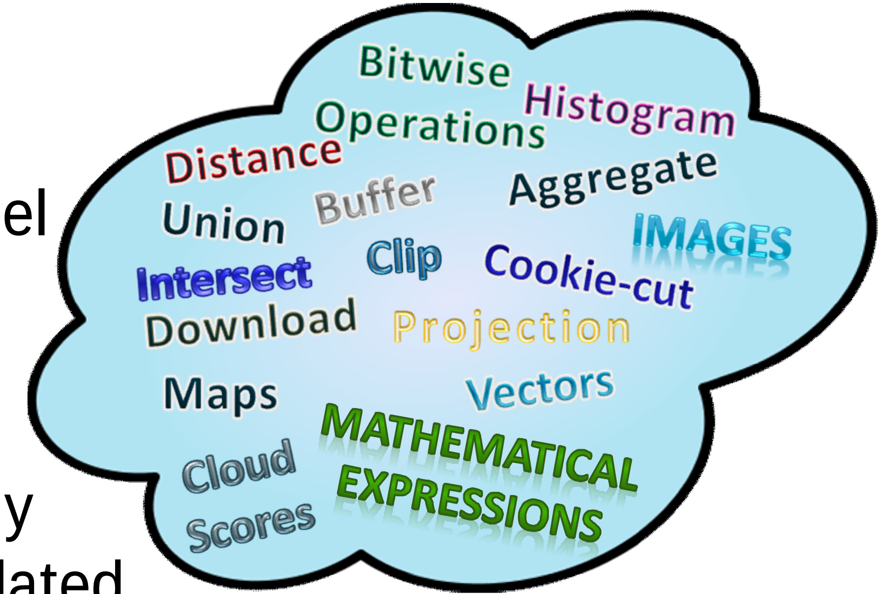
System Architecture Overview



Google Earth Engine

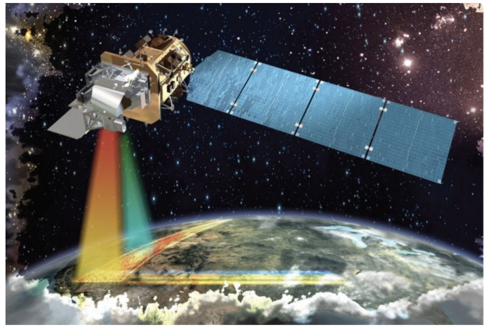
1

- Not to be confused with Google Earth
- Develop and run algorithms on large satellite imagery archives (Landsat, MODIS, etc)
- Access web services via restful API
  - Python
  - Javascript
- Runs in real-time on Google's parallel processing platform
- Just-In-Time distributed model
- Ideal for IrriSAT
  - Define a field and analyse instantly
  - Data archive constantly being updated
  - No need to manage any data

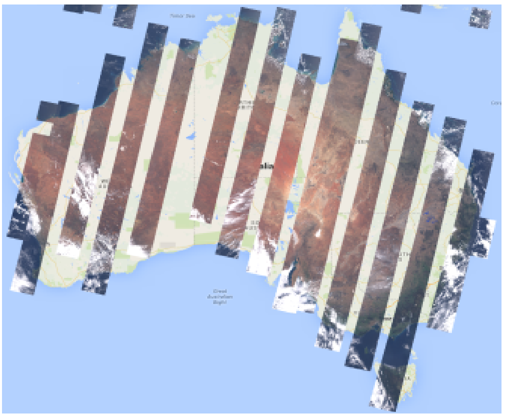


Our approach for determining Kc

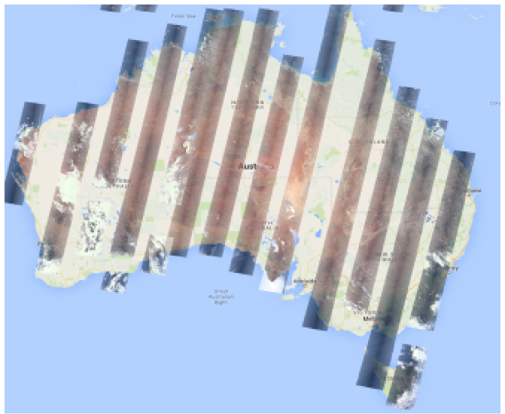
1



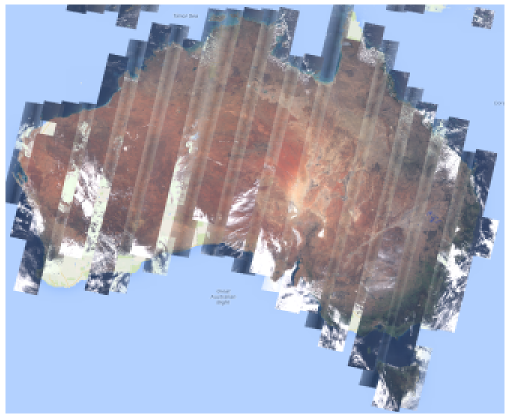
- Landsat Sources 30 m (operational missions)
  - Landsat 8 OLI
  - Landsat 7 ETM+
- 16 Days to orbit earth
- LS8 and LS7 offset 8 days from each other
- Combining provides full coverage every 8 days



Landsat 7



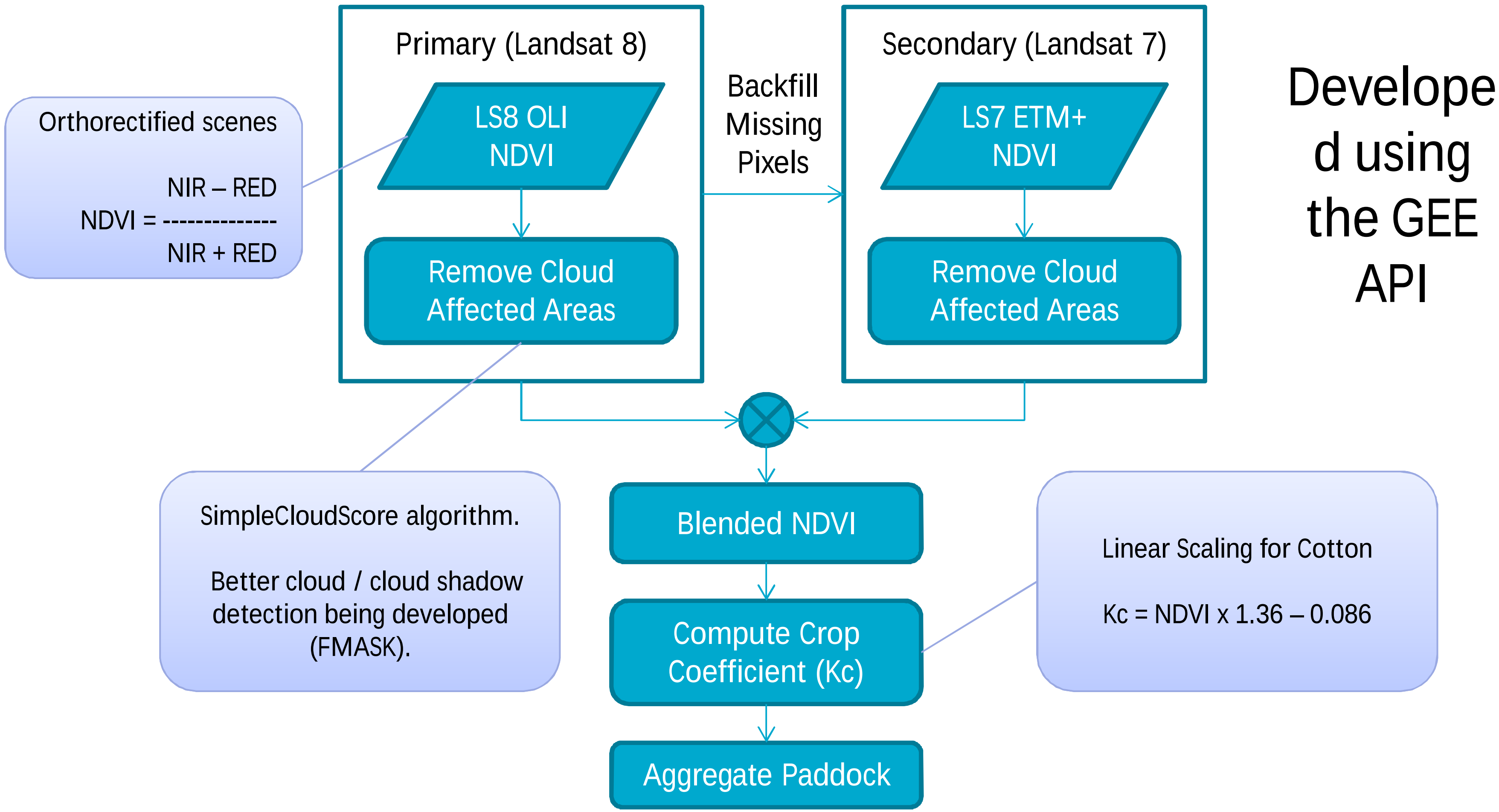
Landsat 8



Combined

Computing Kc (the workflow)

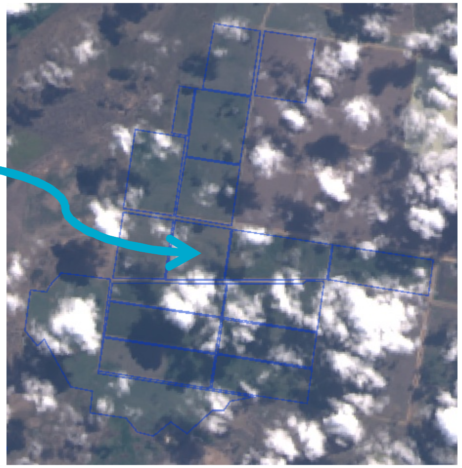
1



Computing Kc (spatial representation)

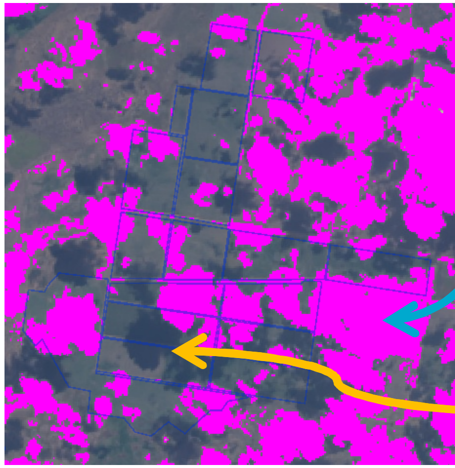
1

Define fields

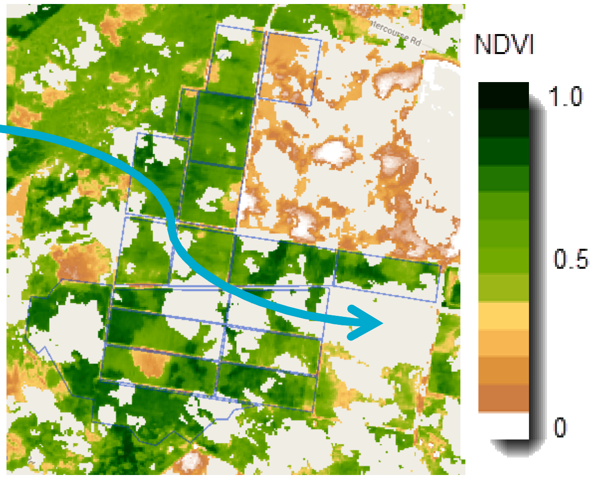


Detect where clouds are present

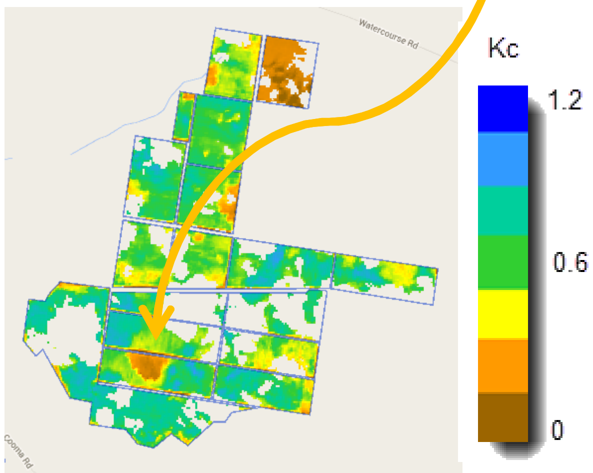
Need to address cloud shadows



Remove pixels from NDVI



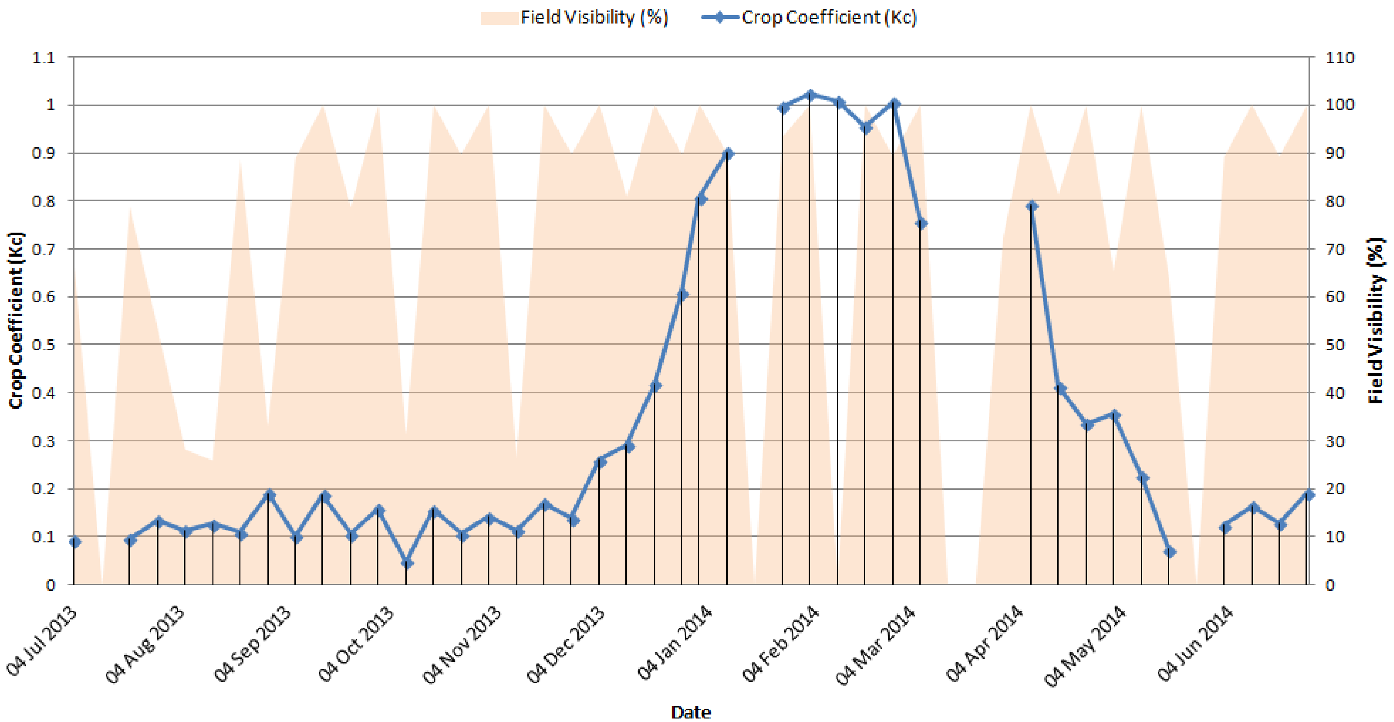
Compute Kc



Generating time series data for a field

1

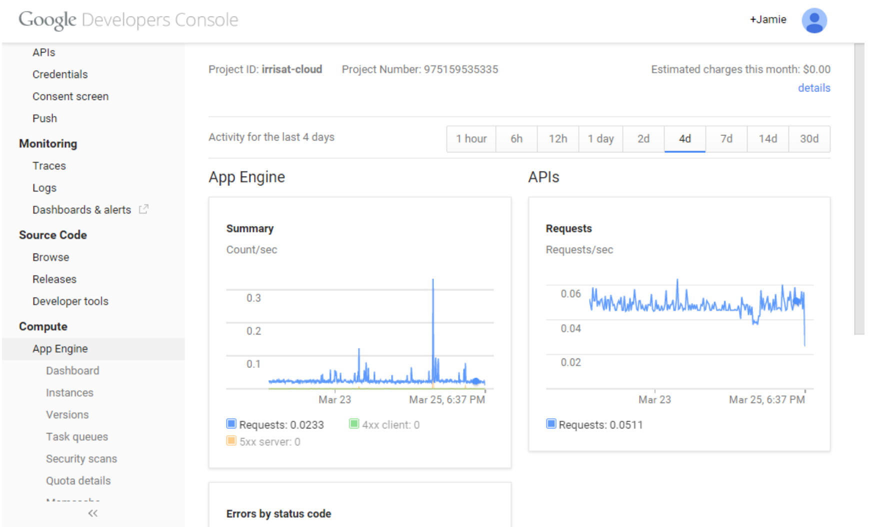
- Spatial algorithm can be reused in time domain
- Aggregation of pixels over a field (ie Kc, Field visibility)
- Again, execution occurs in real-time on Google's servers



Google App Engine

2

- Cloud computing platform we are using to host IrriSAT
  - Scalable – Create new app instances when heavy loads occur
  - Cheap – free so far
- Platform as a Service (PlaaS)
  - App managed via a web console
  - No need to worry about operating system specifics, DBA's etc
- Develop in: Python, Java, PHP, Go (Python for IrriSAT)
- Simple to transfer to industry if opportunity comes along again



Developer Console

Weather Station Data (ETo)

3

- Sensor Observation Service
  - Standardised Web service interface to
    - Query observation data
    - Sensor metadata
    - Features over the web
  - OGC specification – XML/O&M/GML/SensorML Markups


```xml
<om:result>
  <wml2:MeasurementTimeseries gml:id="timeseries.1">
    <wml2:metadata>
    <wml2:defaultPointMetadata>
    <wml2:point>
      <wml2:MeasurementTVP>
        <wml2:time> 2009-11-12T14:00:00.000Z </wml2:time>
        <wml2:value> 0.0 </wml2:value>
      </wml2:MeasurementTVP>
    </wml2:point>
    <wml2:point>
      <wml2:MeasurementTVP>
        <wml2:time> 2009-11-12T15:00:00.000Z </wml2:time>
        <wml2:value> 0.0 </wml2:value>
      </wml2:MeasurementTVP>
    </wml2:point>
    <wml2:point>
      <wml2:MeasurementTVP>
        <wml2:time> 2009-11-12T16:00:00.000Z </wml2:time>
        <wml2:value> 0.0 </wml2:value>
      </wml2:MeasurementTVP>
    </wml2:point>
```


GetObservation Response

- 3 Core methods
  - GetCapabilities - offeres operations and endpoints as well as the available sensor data
  - GetObservation - observed values, including their metadata
  - DescribeSensor - provides sensor metadata (location, parameters, etc)

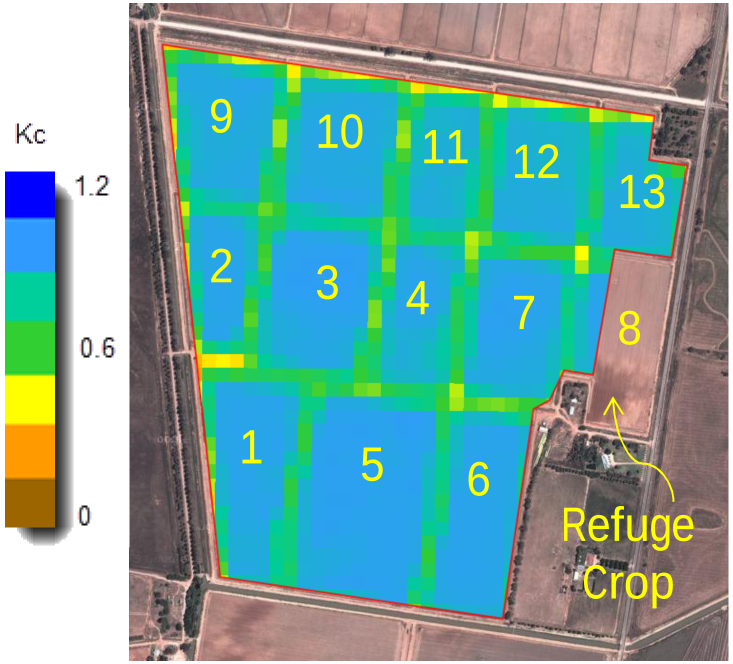
Enable Google Cloud Platform to access Weather Observations within CSIRO

Current Users

- Farmers (around Moree, NSW + southern QLD)
- Consultants (HMAg, Sundown Pastoral Company)
- DPI – Studying water use efficiency between different irrigation setups (spatial variation)
  - Spray
  - Drip, etc
- Software Engineers – building other products on top of IrriSAT API

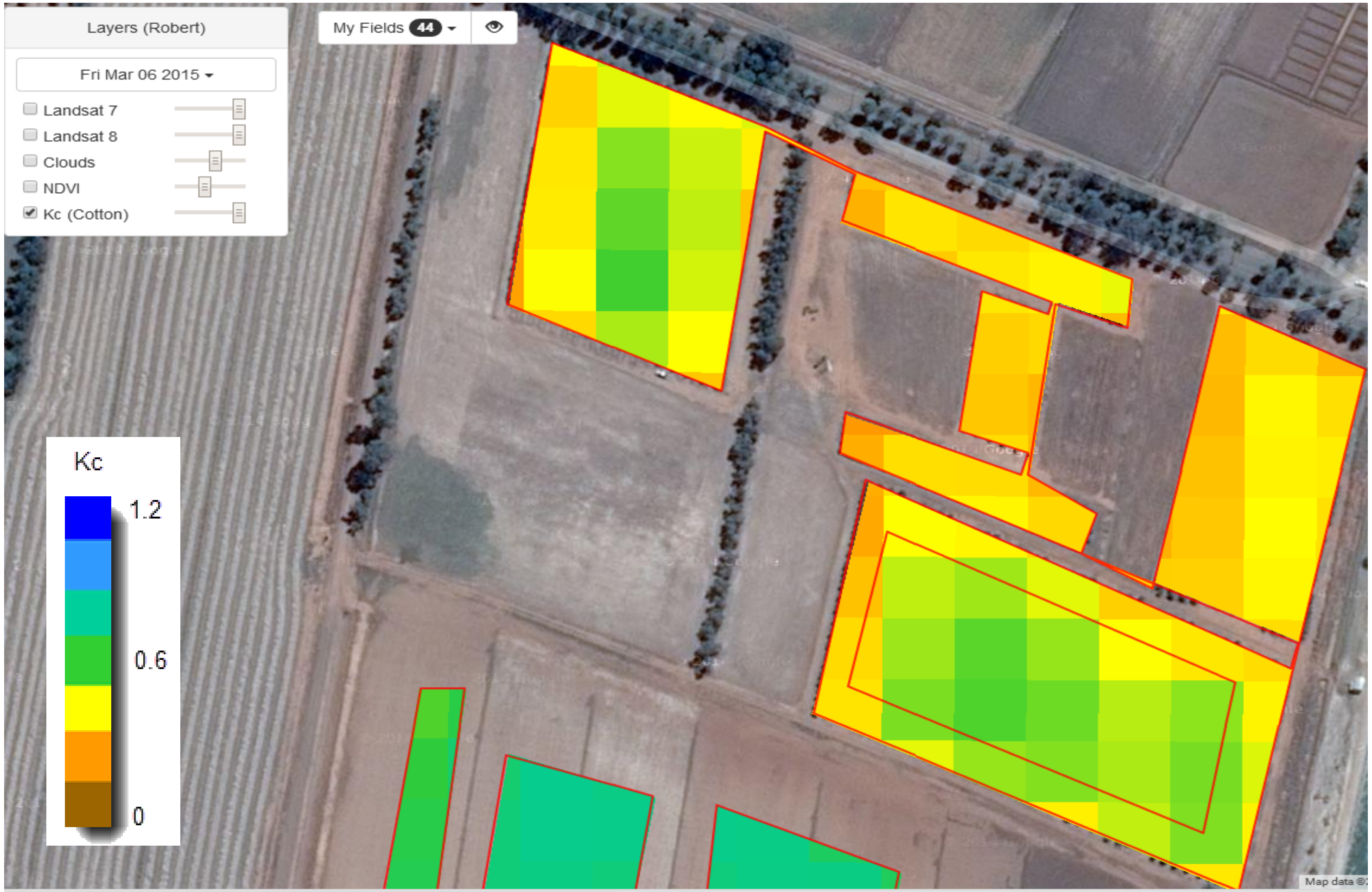
Future challenges

- Complex irrigation regimes
  - Refine the accuracy and ability of the system to meet cotton growers' needs
  - Buffering?
  - Multipolygon?
  - Further algorithms to auto detect roads? Etc
- Decision Support Systems
  - Too complex = won't use
  - Too simple = not useful



How users are actually using IrriSAT

"This saved me 4 months of work"...



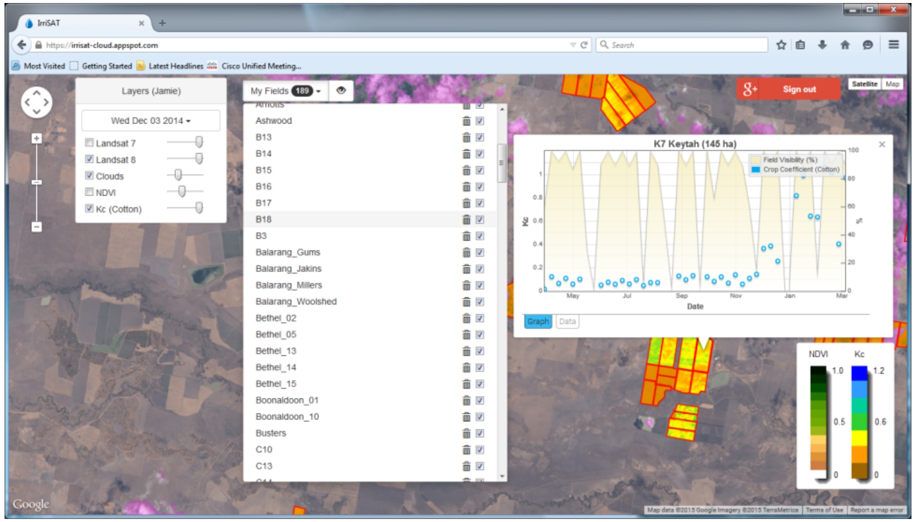
Conclusions

- IrriSAT will provide *real time* crop water use at *broad scale* and *low cost*
- Work to date (enabling data services): **ETc = ETo x Kc**

- Future work involves: refining the *accuracy*; incorporating *7 day forecast* of irrigation demand; and also *benchmarking* against nearby fields.

- Visit the IrriSAT website:

www.irrisat-cloud.appspot.com



Thank you

Land and Water Flagship
Jamie Vleeshouwer
Software Engineer

t +61 7 3833 5589
e jamie.vleeshouwer@csiro.au
w www.csiro.au