AI AND DATACUBES:
TOWARDS A HAPPY MARRIAGE

Peter Baumann, Dimitar Misev
Constructor University, Bremen, Germany
dmisev@constructor.university

1. Introduction

- Datacubes are an accepted cornerstone towards Analysis-Ready Data, enabling high-level services on Big Earth Data (Figure 1)

- AI methods provide enhanced insight on vast amounts of data

- We integrate AI model application on datacubes with focus on

  - *user-friendliness* – declarative datacube query language based on Tensor Algebra

  - *scalability* – transparently use PB of datacubes

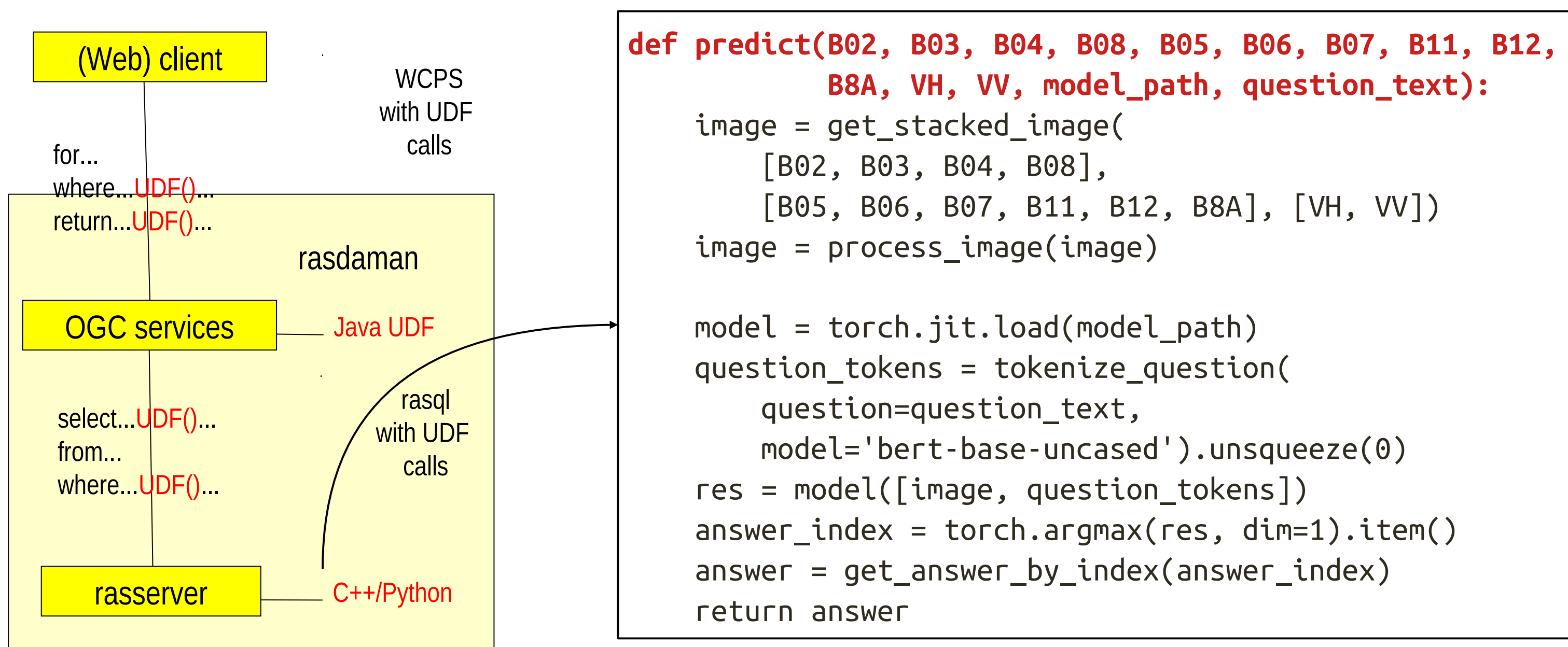 - *flexibility* – model-agnostic method via Python UDFs


```
def predict(B02, B03, B04, B08, B05, B06, B07, B11, B12,
            B8A, VH, VV, model_path, question_text):
    image = get_stacked_image(
        [B02, B03, B04, B08],
        [B05, B06, B07, B11, B12, B8A], [VH, VV])
    image = process_image(image)

    model = torch.jit.load(model_path)
    question_tokens = tokenize_question(
        question=question_text,
        model='bert-base-uncased').unsqueeze(0)
    res = model([image, question_tokens])
    answer_index = torch.argmax(res, dim=1).item()
    answer = get_answer_by_index(answer_index)
    return answer
```

Fig. 2: Architecture of the rasdaman datacube DBMS with UDFs

3. Applications

- **Natural Language Processing (Fig. 3)**
  using Visual Question Answering for Remote Sensing Data (RSVQA)

```
for $B02 in (S2_L2A_B02_10m), $B03 in (S2_L2A_B03_10m),
    $B04 in (S2_L2A_B04_10m), $B08 in (S2_L2A_B08_10m),
    $B05 in (S2_L2A_B05_20m), $B06 in (S2_L2A_B06_20m),
    $B07 in (S2_L2A_B07_20m), $B11 in (S2_L2A_B11_20m),
    $B12 in (S2_L2A_B12_20m), $B8A in (S2_L2A_B8A_20m),
    $VH in (S1_GRDH_IW_DV_VH), $VV in (S1_GRDH_IW_DV_VV)

let subs2 := [ansi("2023-06-08"), Lat(53.0450:53.0570), Lon(8.7800:8.7920)],
    subs1 := [ansi("2023-06-04"), Lat(53.0450:53.0570), Lon(8.7800:8.7920)]

return rsvqa.predict($B02[subs2], $B03[subs2], $B04[subs2], $B08[subs2],
                     $B05[subs2], $B06[subs2], $B07[subs2], $B11[subs2],
                     $B12[subs2], $B8A[subs2], $VH[subs1], $VV[subs1],
                     "rsvqa_trained_model.pt",
                     "Are agricultural areas or wetlands present?")
```

- **Crop Classification (Fig. 4)**

```
for $s2 in ( sentinel2_flevopolder_10m_7x4bands )

let aoi := [E(677400:679900), N(5834260:5836760)]

return encode(
           fairicube.predictCropClass( $s2[aoi], max($s2[aoi]) ),
       "png")
```
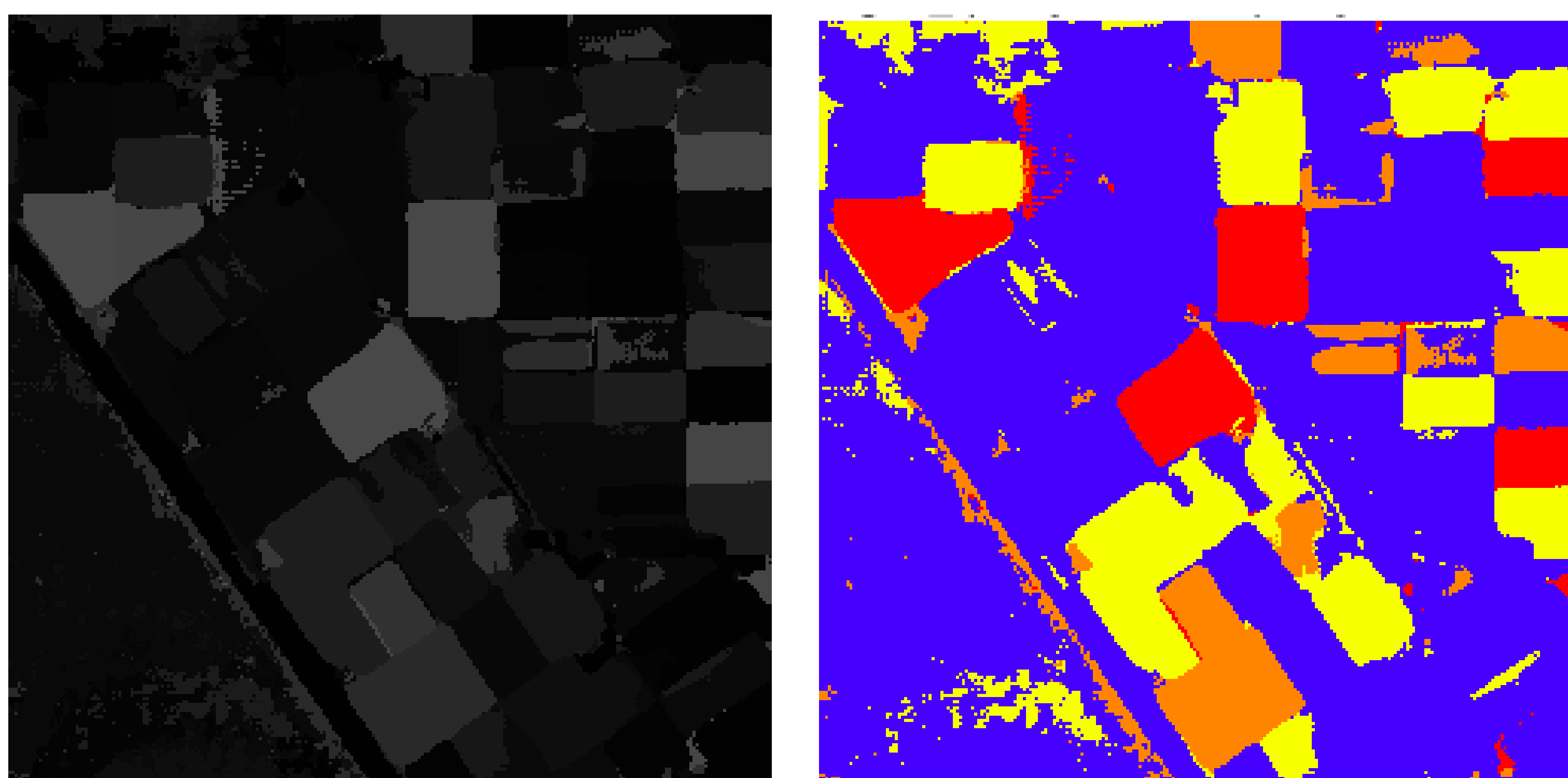

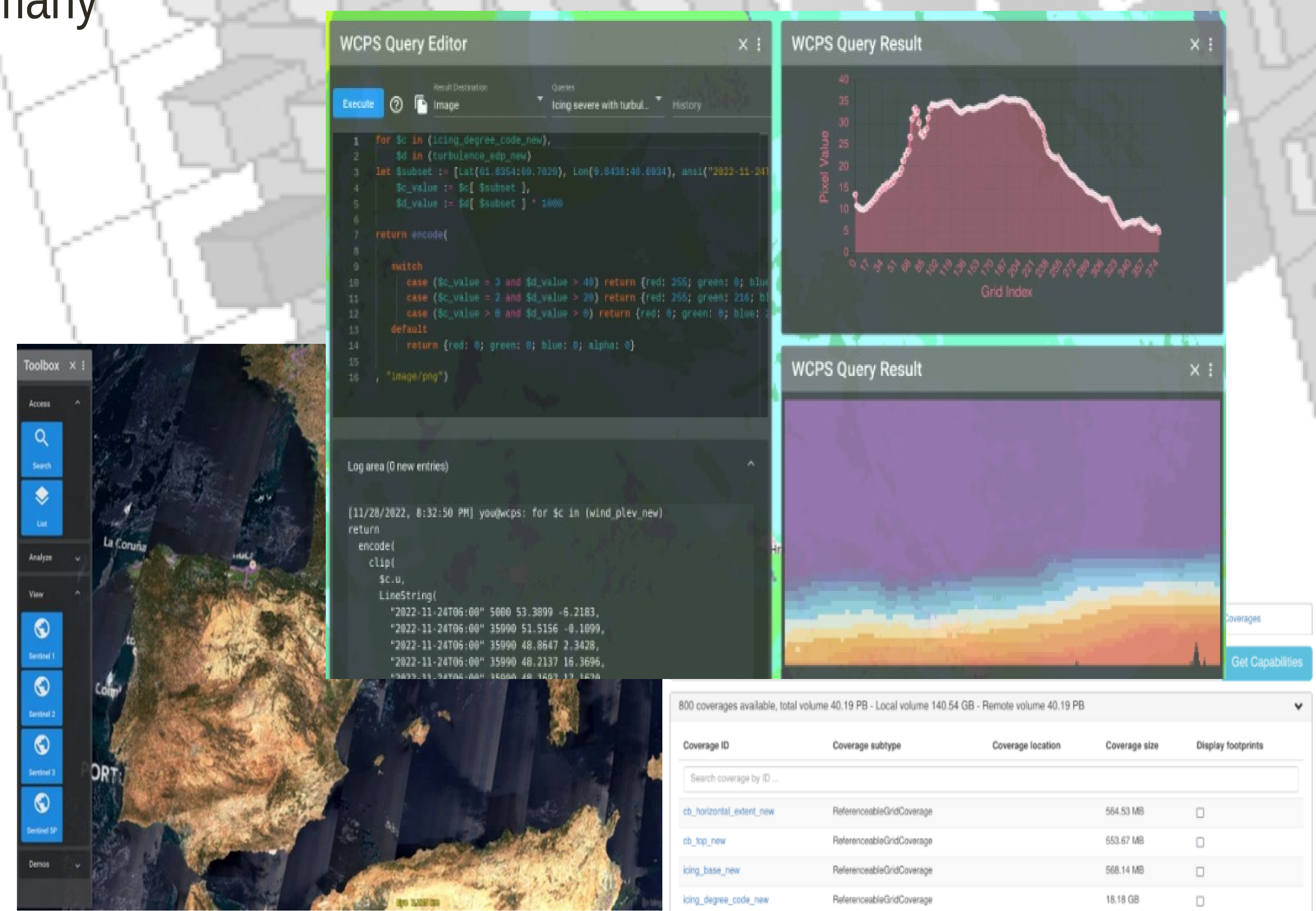*Fig. 4: Crop classification query results (FAIRiCUBE project [2])*

Fig. 1: High-level Datacube services: WMS/WMTS, WCPS, WCS

2. User-Defined Functions

- Extend query language on datacubes with new functions

- High performance *Java*, *C++* or *Python* extensions, directly embedded in the query processing pipeline

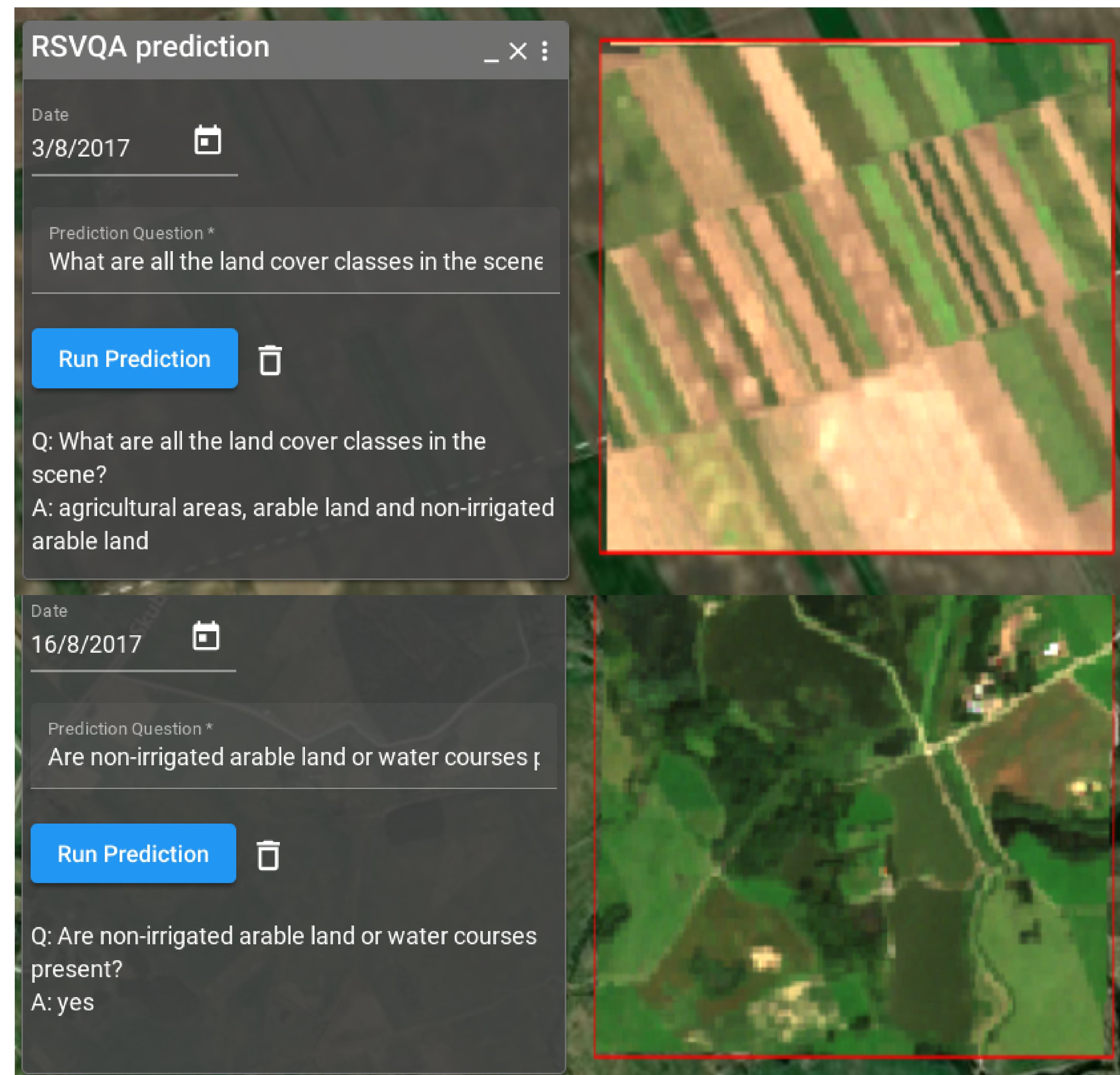- WCPS (geo-datacubes) and rasql (domain-neutral arrays)

Fig. 3: "Chatting" about remote sensing data (AI-Cube project [1])

4. Conclusion and Outlook

- Seamless integration of AI/ML prediction on datacubes through UDFs

- *Open questions*

  - User-friendly model management

  - Guidance on model applicability

  - Integration of AI/ML training

- Live demo: https://aicube.rasdaman.com/rasdaman-dashboard/