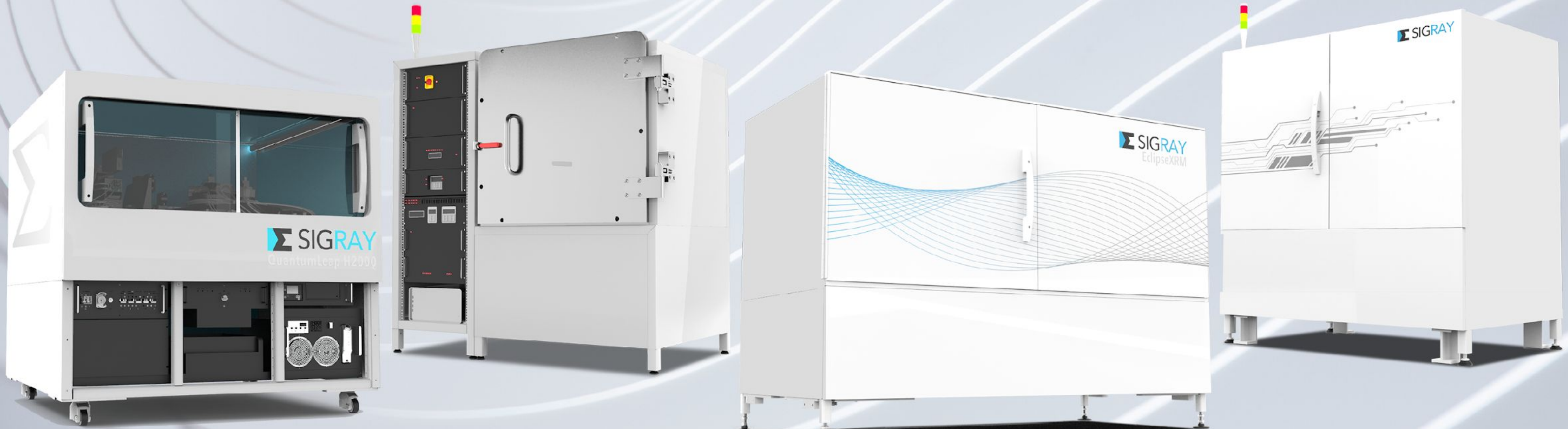
# EPICS use in Industry- Sigray Inc.

Presenter: Benjamin Stripe
Co-Authors: Hong Truong, Ibrahim Saleh, Richard Farnsworth, Wenbing Yun



## Founded in 2013

- Dr. Wenbing Yun (OSA Fellow and serial entrepreneur that founded Xradia, now Carl Zeiss X-ray Microscopy) and Sylvia Lewis

## Our Technology:

- Strong IP: 64 patents, 30+ pending, many trade secrets
- Disruptive x-ray components (source & optics)
- 5 world leading product families

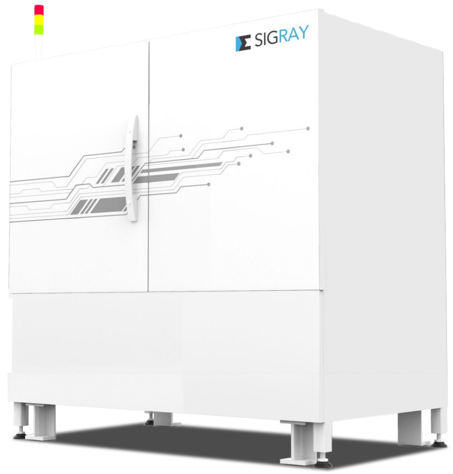
## Rapidly Growing:

- 34k sq. ft. facility in Concord, CA (San Francisco Bay Area) and 82 employees
- Global installation base of leading universities and companies (semiconductor & pharma)

# Intro to Sigray

Mission: Bring next-generation x-ray analytical capabilities from the synchrotron to the laboratory

# Suite of Synchrotron Beamline-like Systems

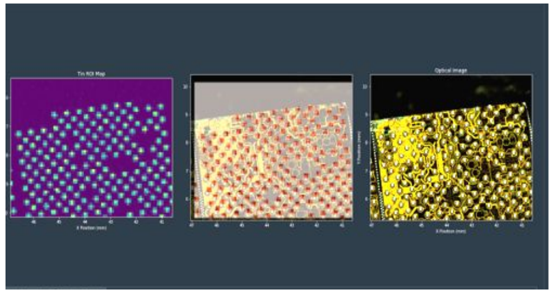


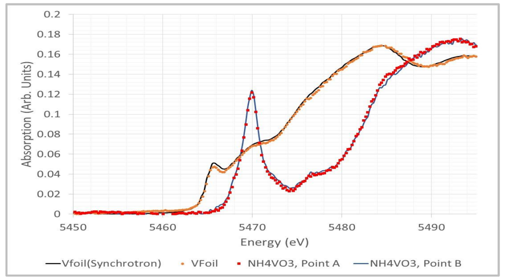
**AttoMap microXRF**
Highest sensitivity to low Z
Highest resolution microXRF

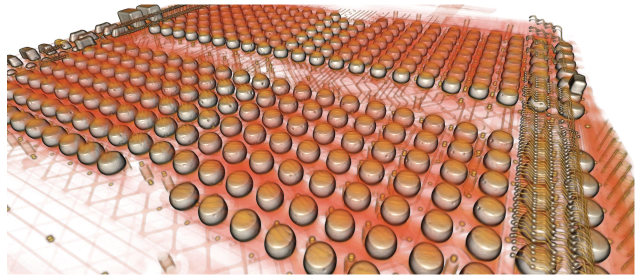
**QuantumLeap XAS**
Only commercial f-XAS
Highest quality EXAFS

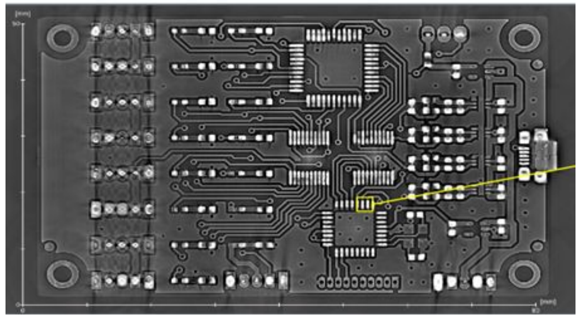
**EclipseXRM**
Highest resolution (0.3µm spatial) 3D x-ray microscope on the market

**Apex XCT**
Highest throughput 3D x-ray tool (>1000X) for semi





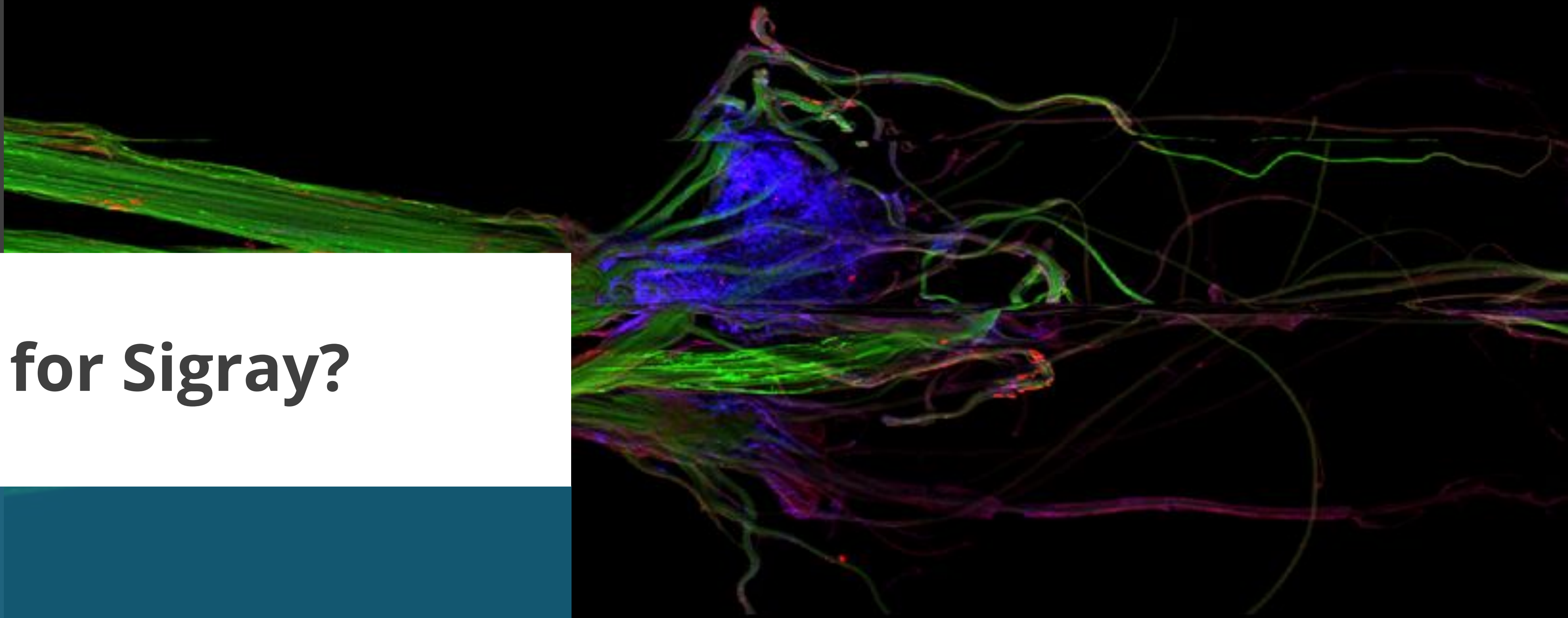




- Introduction to Sigray (Complete)
- Why EPICS for Sigray?
- Scope and deployment differences from larger laboratory deployments
- Solutions we have implemented
  - Docker, Portainer, CIDER - internally developed deployment management
- Outstanding and Emergent Issues
  - Instrument API and automation access.
  - Instrument state tracking, Error handling and recover.
- “Sigray Connect” - GRPC instrument state machine

## Overview

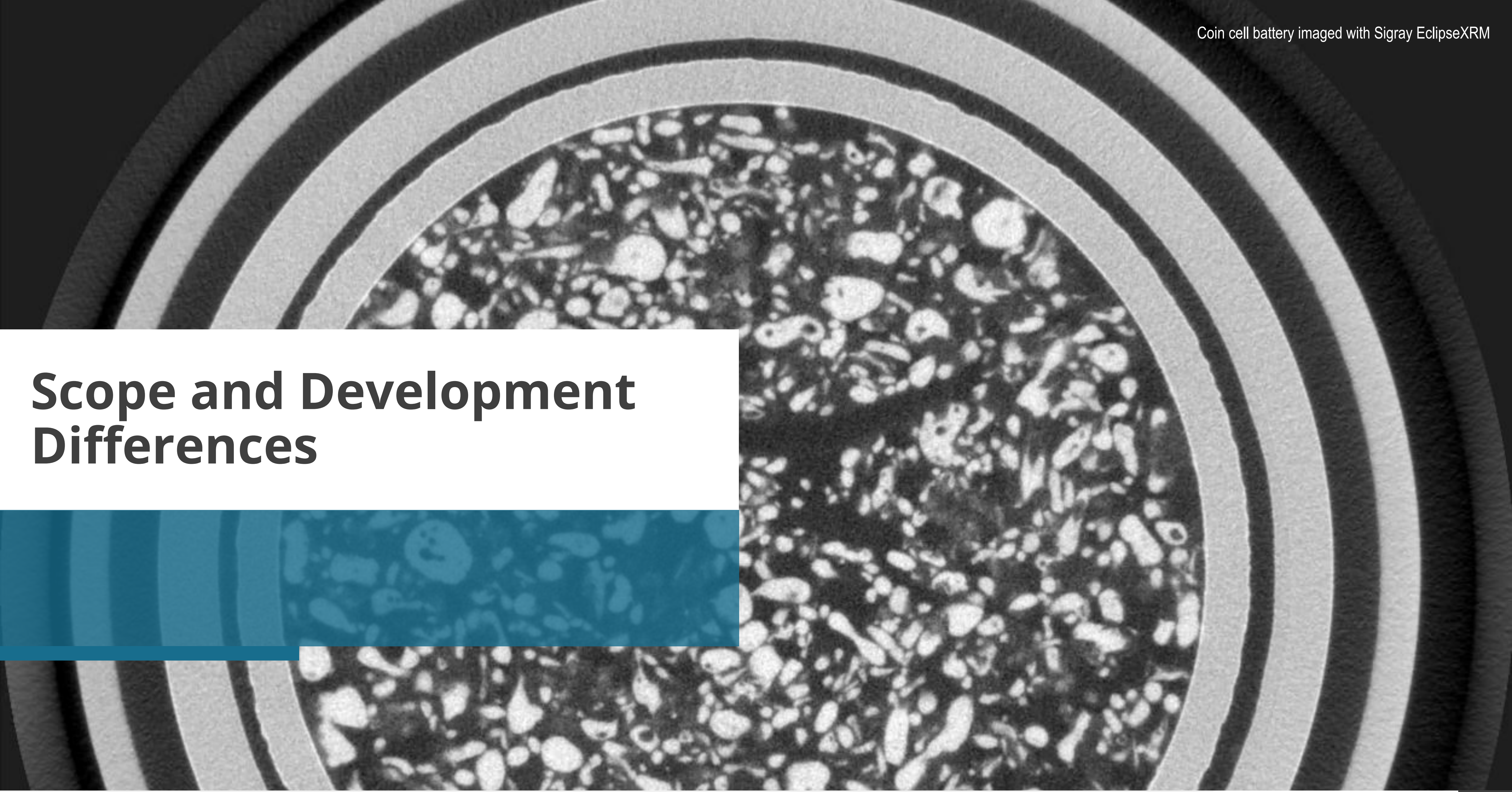
## Why EPICS for Sigray?


Control turf grass imaged on Sigray AttoMap. Potassium (green), calcium (red), manganese (blue) at 20 micron step sizes.

## Why EPICS for Sigray

### Historical Choices

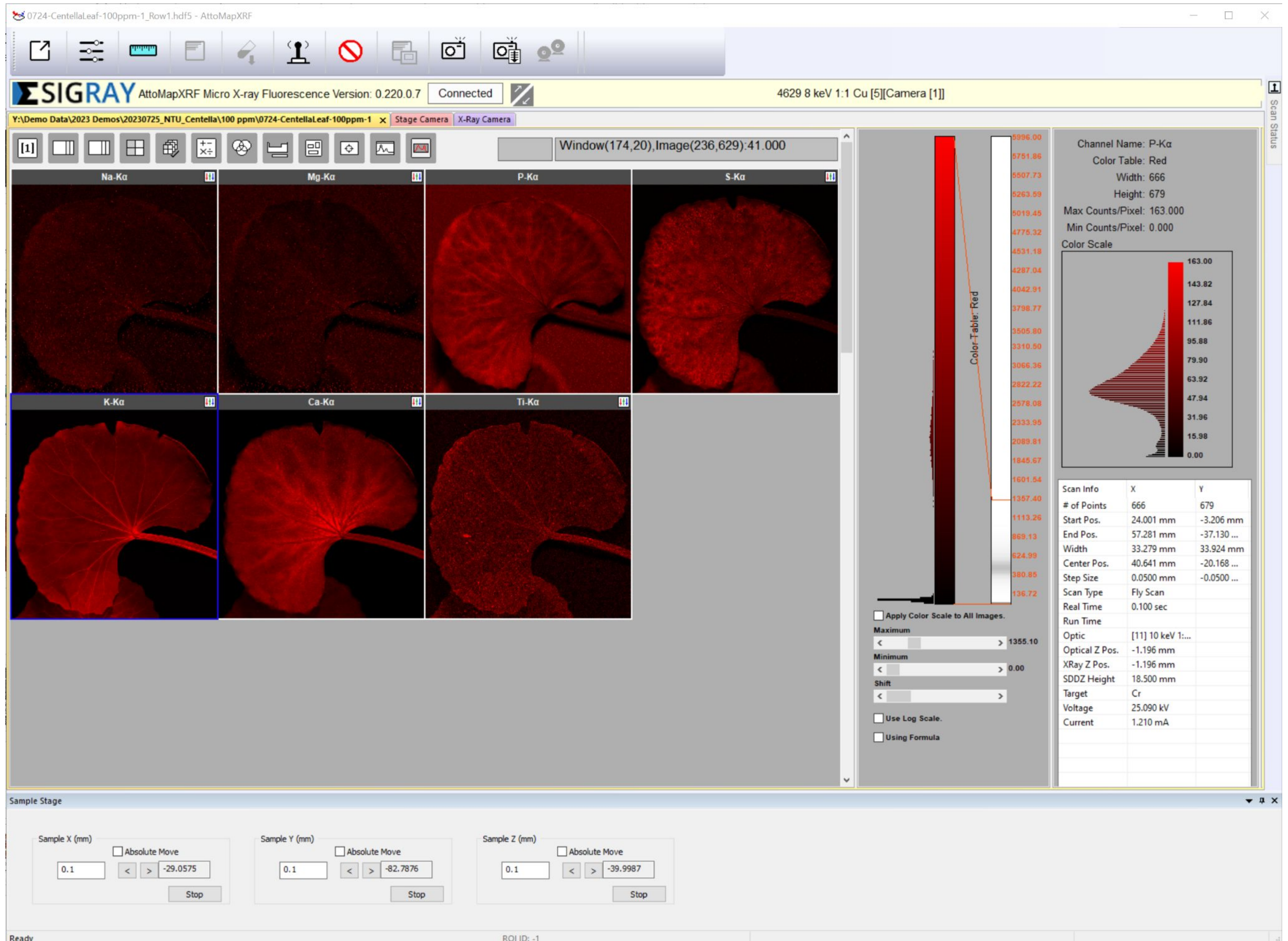
- Sigray's first instrument was a scanning x-ray fluorescence system (XRF)
- System requirements
    - 15 motors
    - Alignment of x-ray capillary optic
        - Downstream Scientific CCD
    - Sample Alignment - Real time on-axis camera
    - Data acquisition from Silicon Drift Detector.
    - Step and Fly scanning integration
- For people with experience from the synchrotron light source community, there are some obvious options for integration spanning many choices of motion control, pulse processors and CCDs
    - EPICS
    - TANGO
    - Labview
    - Roll your own
    - etc ...


Coin cell battery imaged with Sigray EclipseXRM

## Scope and Development Differences

## Scope and development differences

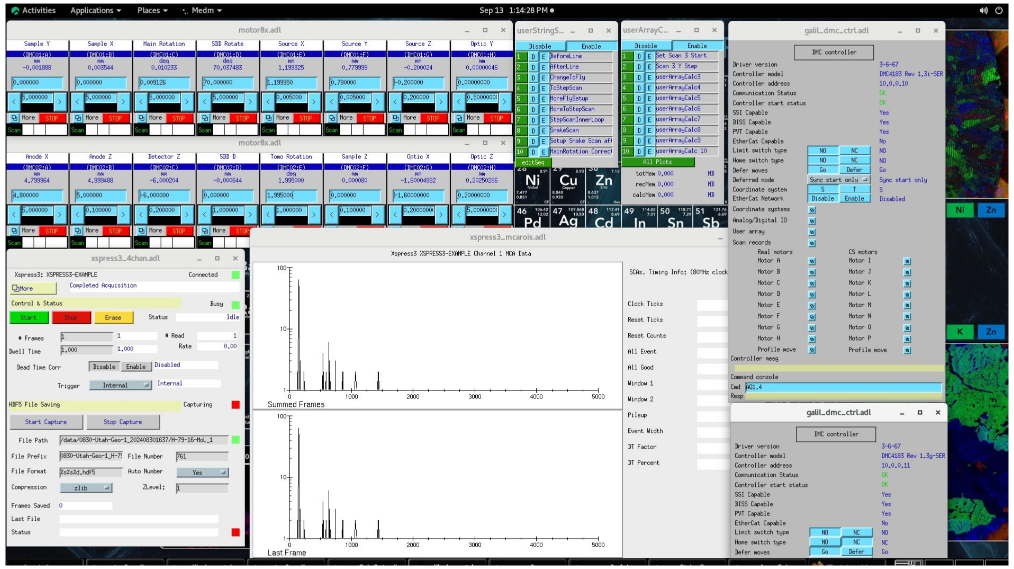
- As an industrial deployment, very few users actually see the underlying controls system
- We have developed top level GUI software for driving most of the machines.


0724-CentellaLeaf-100ppm-1_Row1.hdf5 - AttoMapXRF
SIGRAY AttoMapXRF Micro X-ray Fluorescence Version: 0.220.0.7
Connected
4629 8 keV 1:1 Cu [5][Camera [1]]
Y:\Demo Data\2023 Demos\20230725_NTU_Centella\100 ppm\0724-CentellaLeaf-100ppm-1
Stage Camera
X-Ray Camera
Window(174,20),Image(236,629):41.000
Na-Kα
Mg-Kα
P-Kα
S-Kα
K-Kα
Ca-Kα
Ti-Kα
Color-Table: Red
Apply Color Scale to All Images.
Maximum
1355.10
Minimum
0.00
Shift
Use Log Scale.
Using Formula
Scan Status
Sample Stage
Ready
ROI ID: -1

Channel Name: P-Kα
Color Table: Red
Width: 666
Height: 679
Max Counts/Pixel: 163.000
Min Counts/Pixel: 0.000
Color Scale

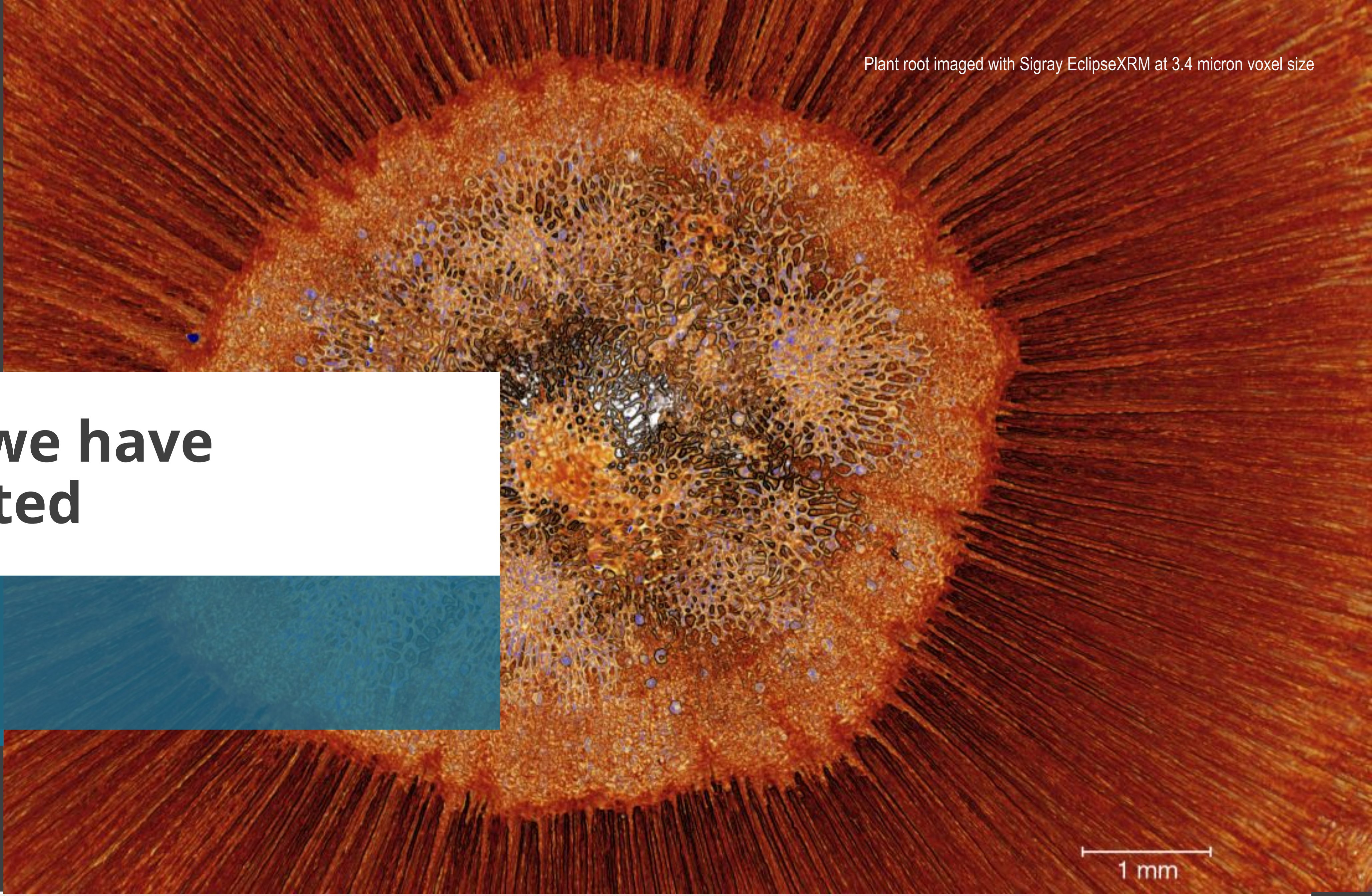
| Scan Info | X | Y |
|---|---|---|
| # of Points | 666 | 679 |
| Start Pos. | 24.001 mm | -3.206 mm |
| End Pos. | 57.281 mm | -37.130 ... |
| Width | 33.279 mm | 33.924 mm |
| Center Pos. | 40.641 mm | -20.168 ... |
| Step Size | 0.0500 mm | -0.0500 ... |
| Scan Type | Fly Scan | |
| Real Time | 0.100 sec | |
| Run Time | | |
| Optic | [11] 10 keV 1:... | |
| Optical Z Pos. | -1.196 mm | |
| XRay Z Pos. | -1.196 mm | |
| SDDZ Height | 18.500 mm | |
| Target | Cr | |
| Voltage | 25.090 kV | |
| Current | 1.210 mA | |

| Sample X (mm) | Sample Y (mm) | Sample Z (mm) |
|---|---|---|
| Absolute Move | Absolute Move | Absolute Move |
| 0.1 / -29.0575 | 0.1 / -82.7876 | 0.1 / -39.9987 |
| Stop | Stop | Stop |



## Scope and development differences

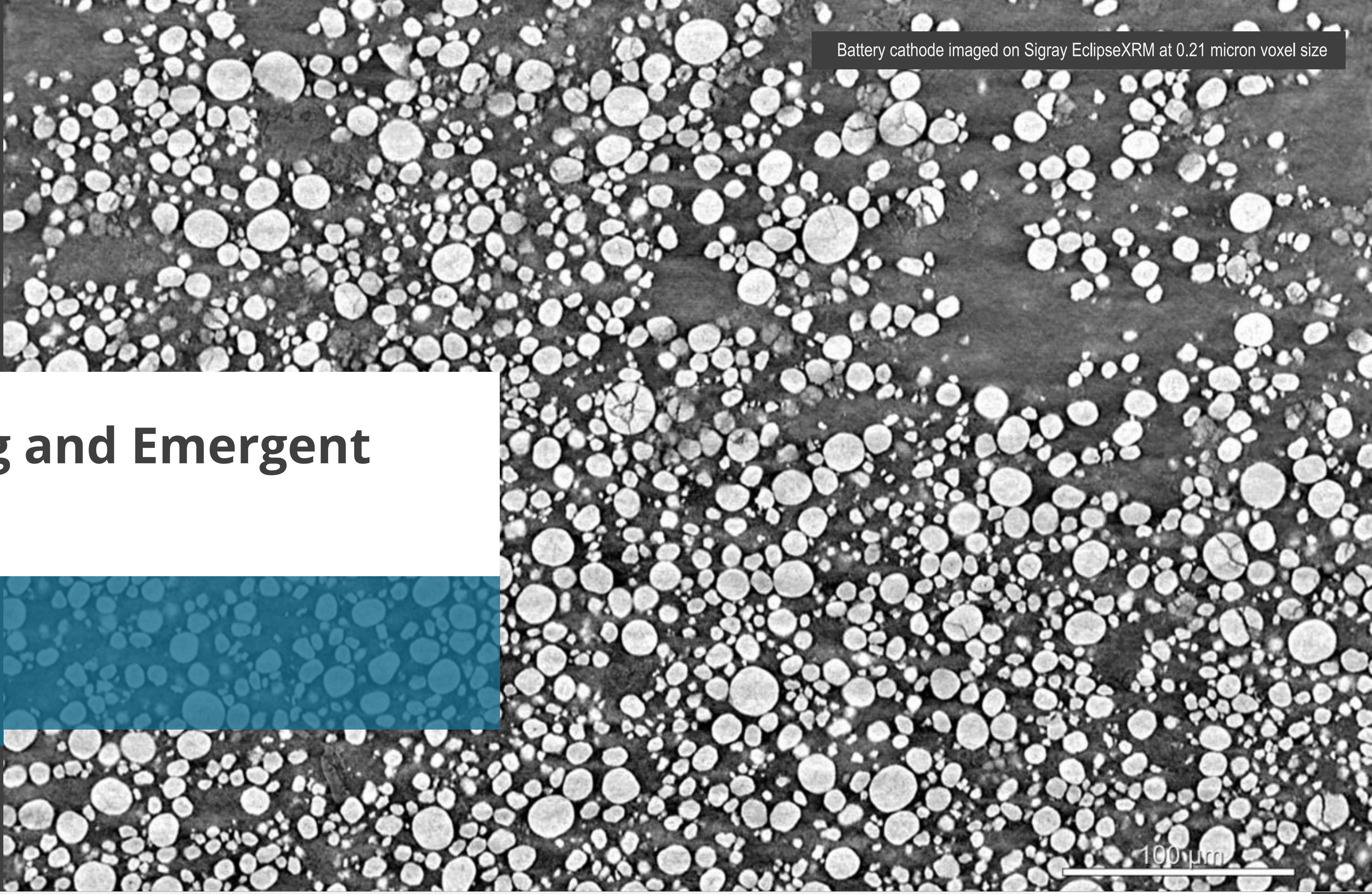
- Top level software adds a lot of features
  - Starting Acquisitions from optical images
  - Automating alignment
  - Batch scans
  - High level sequencing and automation
- MEDM is still regularly used as an engineering and Rapid R&D interface!
  - EPICS has a real strength here! Software R&D time for complex motion and acquisition can be very slow if done using a self-made or in-house framework
- Problems arise when you have 5 different systems
  - They develop incremental changes with time (Newer bigger detectors, faster motors ...)
  - You need to deploy update and maintain the controls software on machines in concrete basements with **no internet access!**


Plant root imaged with Sigray EclipseXRM at 3.4 micron voxel size
1 mm

## Solutions we have Implemented

- Like others in the community we looked at packaging our IOCs as application deployment Images
  - Settled on docker containers
- Still have a number of variations from system to system
  - Different detector IOCs to run
  - Different pixel defect maps
  - Encoder and stage / motion controller changes
  - Developed a tool called CIDER to help with this
    - Use a base Docker Image
    - Allows us to track and mountain certain files at build or run time as needed
      - st.cmd changes
      - .db alterations
      - Pixel defect maps

See Hong Truong’s presentation later for the specifics


Battery cathode imaged on Sigray EclipseXRM at 0.21 micron voxel size

## Outstanding and Emergent Issues

SIGRAY

## Outstanding and Emergent Issues

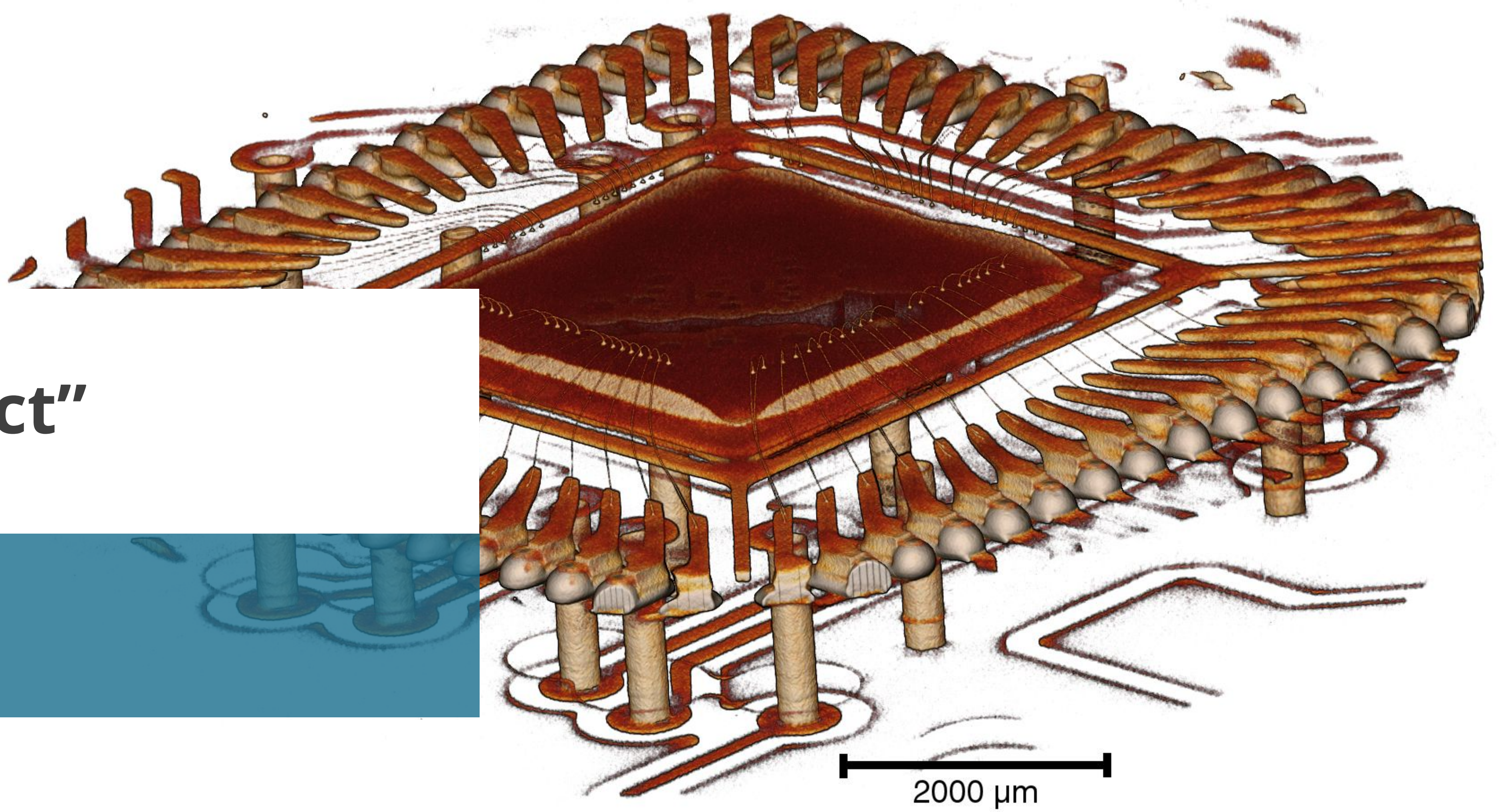
- Because of our historical choices and being a small company,  as we added products,  we tasked different programers to develop user interfaces for each system
  - User interfaces in MFC, Python-QT , UWP, some C# - QT
- Most of these apps direct control logical sequencing for the instruments
  - Turn on x-rays
  - Move motors directly through channel access
  - Acquire and render overview images
  - Set up scan parameters, populate the scan record, and render data

## Outstanding and Emergent Issues

This had lead us to three major emergent problems as we grow and look to support all of these systems

- There is very little reuse of code between systems at the user interface level
    - Simple things like turning on X-rays can be run through EPICS, through a stand alone application, or directly through the acquisition GUI.
- Systems “Statedness”
    - Systems runs with a workflow oriented GUI, sequences things like scan record as needed and monitors until the process finishes
    - Dealing with interrupts, errors, power outages.
- No higher level interface: API or interface to automate the system
    - Move in to industries like semiconductor come with wants to
        - Control the tool as part of a larger system
        - Automate process control
        - CAD Based feature identification

Component on Intact PCB imaged with Sigray Apex XCT at 1.5 micron



# “Sigray Connect”

EPICS CA and GRPC glued

SIGRAY

## Sigray Connect

### Work in progress

- Essentially a modular state machine based middle man
    - Use Google Remote Procedure Calls (gRPC)
        - gRPC is supported in basically every language
        - Can call a procedure in python that runs on another machine in C# or any other language
        - Is a protocol that can be versioned and is fault tolerant
        - Asynchronous server style communication
    - Allows us to Implement system "Statedness"
        - Allows us to address aborts, interrupts and errors
    - Allows us to reuse procedures between systems
        - No need to write two CAD software interface modules
    - Allows us to separate procedures on different machines
        - Start Scan procedure can stay on the controls machine
        - Updating the Acquisition client no longer runs a risk of changing controls code. Similar to implementing firmware  procedure calls, even if if the scan sequence is different as machine Hardware evolves the interface does not care.

## Sigray Connect

### Work in progress

- This interface allows us to do things like trigger the exact MFC scan start sequence we have implemented in the in acquisition GUI through gRPC in a python script!
    - Previously system scripting/R&D/Automation using python required setting up the scan records directly. A completely separate scan start sequence. This improves code and system robustness significantly!
- Sigray Connect is Modular in the sense that.
    - We can add interface modules for that manage the state of Motion controllers, area detector, turbo pumps, enclosure lighting....
    - We can add modules to create other communication channels to interact with the state machine
        - HTTP based process command API for semiconductor
        - Could support REST or even RS232 API access....
    - Modules are introspective
        - Can automatically generate simple engineer level GUI screens to link inputs and outputs. (Much like how we use MEDM it's really powerful to have an engineering interface for rapid R&D)
    - While most of the modulus interact with EPICS via Channel Access, there is no reason we have to restrict ourselves in this way. With the developments in modern APIs some CDDs for example, can be much quicker to develop directly.

## Discussion

Some of these are issues standard in software development

- Code base splitting
- Multiple languages
- Historical baggage

These issues are our own making. Have answers relatively independent of the community

Other Issues are current to the community

- “Statedness”
- Code/function/procedure reuse
- Higher level Asynchronous control and automation

Projects like Bluesky with its server modules look to overcome many of the same obstacles. Although still relatively driven be python command line calls.

In my opinion Channel Access and EPICS base enable system controls at scalability,speed, and functionality that really is amazing.

At the code and deployment level it is in sharp need of a modern build environment. Even an upgrade to apt-get level deployment would be amazing. Currently too common to see dockers/images as management solutions.

This would also enable much more direct modernization of high level controls interfaces and or reusable modules like Bluesky. Ways to make reusable acquisition routines with abort,pause,stop….