Sietch
hardware database system

Nathaniel Tagg
ntagg@otterbein.edu
Otterbein University
Aug 6 2020
FD Technical Board Meeting

Sietch is a tool designed to

easily and flexibly track components and tests in the APA or elsewhere.

It's a **web application.** Built to work well on mobile as well as desktop.

You can start playing with it now!

Development server http://dev.sietch.xyz
"Production server" https://sietch.xyz
(No data yet)

Technical details: Connects uses MongoDB database backend. Based on open source platform: Node.js, FormIO.js, Express, Passport. Runs in OSX or any Linux. Uses auth0.com for authentication and user management. Hosted on my own VM at Otterbein for now. Backed up daily.

What is it?

Use cases I've had in mind:

- QA/QC data (e.g. wire tensions) that might be relevant to later physics analysis
- Virtual "traveller" documents
- Tracking component relationships (which cards built into which APA)
- Inventory tracking
- Checklists

Design philosophies

Designed to be used by people on the ground.

Make it easy as possible to get data in.

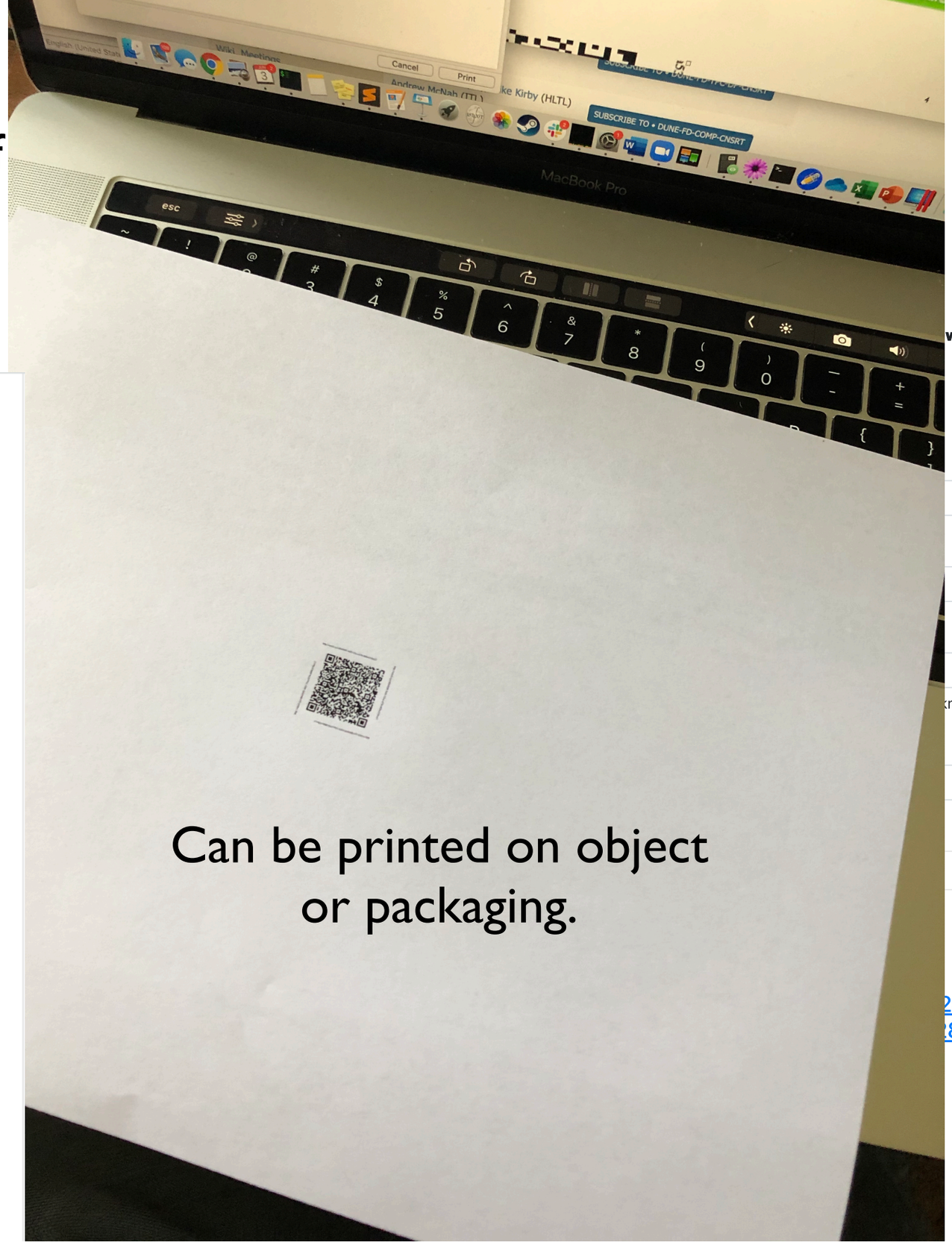
Flexible schemas - allow easy changes as things evolve

Never delete anything. All objects are either versioned or immutable.

Rapid development - we need this yesterday.

Every object has a unique “serial” number in the database (a UUID), represented by a URL in a QR code:

If



Basic concepts

Every object has a unique "serial" number in the database (a UUID), represented by a URL in a QR code:

If you scan this code, you get the "traveller" for this document:

Protodune APA: protoAPA US APA 004

Edit this component

Component UUID
71d9bdc0-ca93-11ea-bcd9-1f445262d4f0

Version 1
edited by Nathaniel Tagg's script on July 20th 2020, 2:15:20 pm

History

Effective as of
July 20th 2020, 2:15:20 pm (16 days ago)

Name	protoAPA US APA 004
APAID	US APA 004
Frame Serial Number	004
Head S N	5-01-01 12:00 AM
Center S N	6-01-01 12:00 AM
Side 1 S N	7-01-01 12:00 AM
Side 2 S N	7-01-01 12:00 AM

Protodune 1 SP APA 4
Translated from Jason Laffin's DB

Basic concepts

Aside… my view is that the best way to deal with component identification is in two parts:

a PBS system to identify TYPES of parts
human readable, well-defined, heriarcical

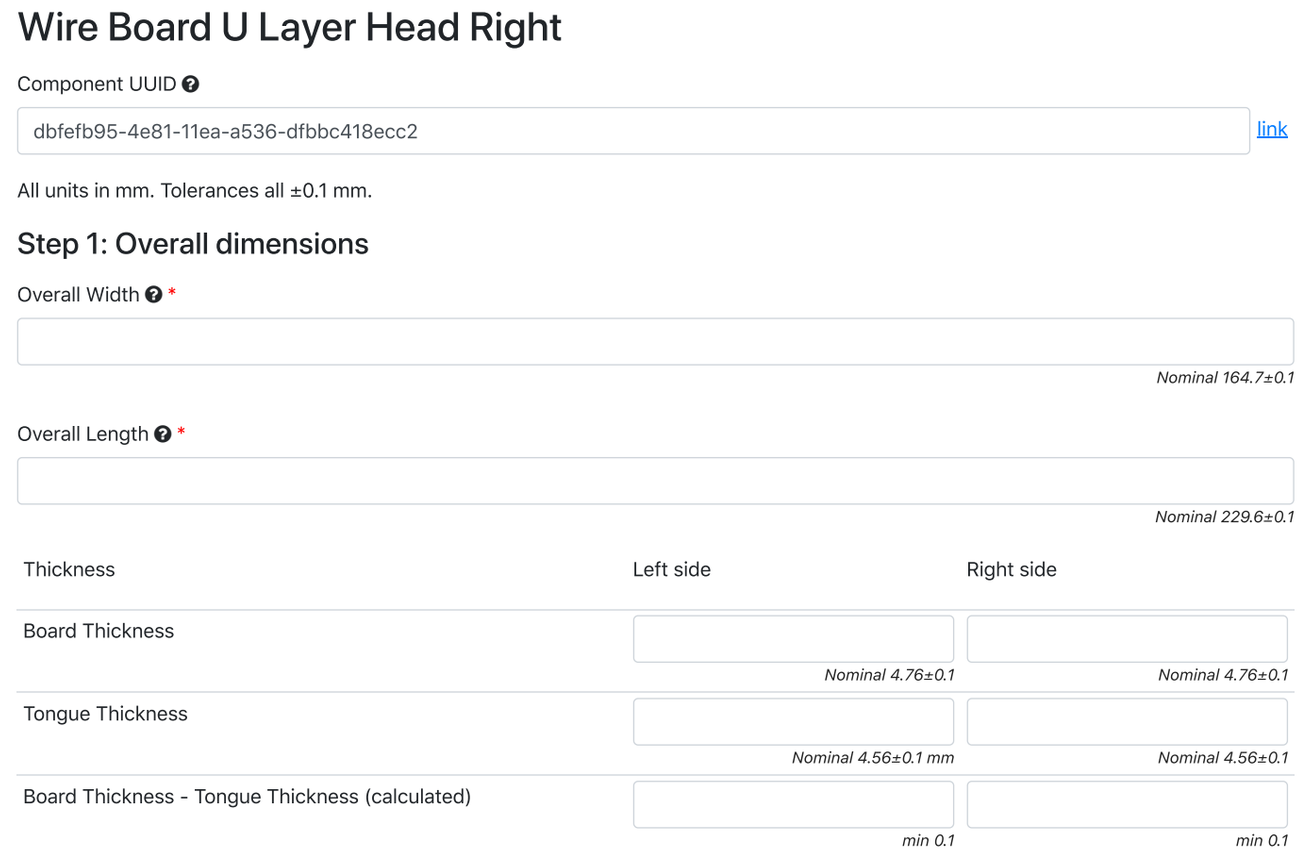
a unique URL to identify specific physical components
linked to virtual travellers, need not be centralized

This lets us get started NOW.

When you enter data about any component, it is automatically:

- associated with that component's UUID number
- tagged with your user ID and email
- timestamped (and optionally GPS stamped)
- entered into the permanent database

Example form for data entry



Wire Board U Layer Head Right

Component UUID

dbfefb95-4e81-11ea-a536-dfbbc418ecc2 link

All units in mm. Tolerances all ±0.1 mm.

Step 1: Overall dimensions

Overall Width *

Nominal 164.7±0.1

Overall Length *

Nominal 229.6±0.1

Thickness	Left side	Right side
Board Thickness	Nominal 4.76±0.1	Nominal 4.76±0.1
Tongue Thickness	Nominal 4.56±0.1 mm	Nominal 4.56±0.1
Board Thickness - Tongue Thickness (calculated)	min 0.1	min 0.1

Step 2: Visual checks

Silkscreen is OK

- Pass
- Fail

Right Side Cut Corner Vertical

Nominal 9.00 mm

Right Side Cut Corner Horizontal

Nominal 8.80 mm

Notch - Vertical

Nominal 9.00 mm

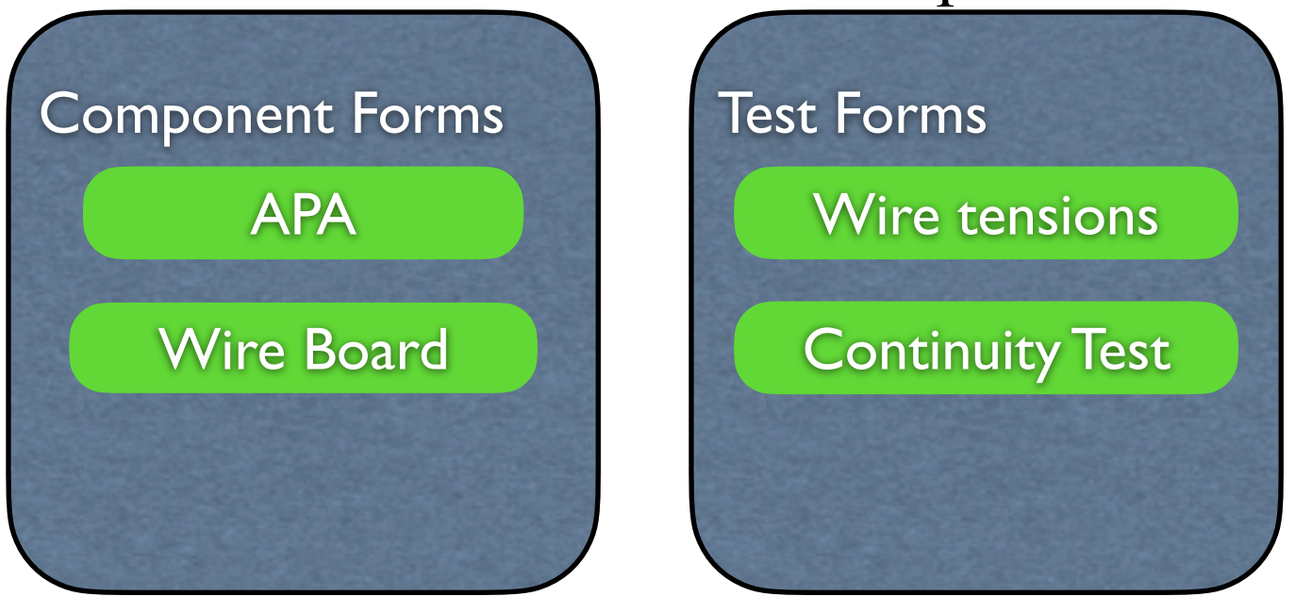
Notch - Horizontal

Schema are flexible!
They can change fluidly without affecting DB behavior.

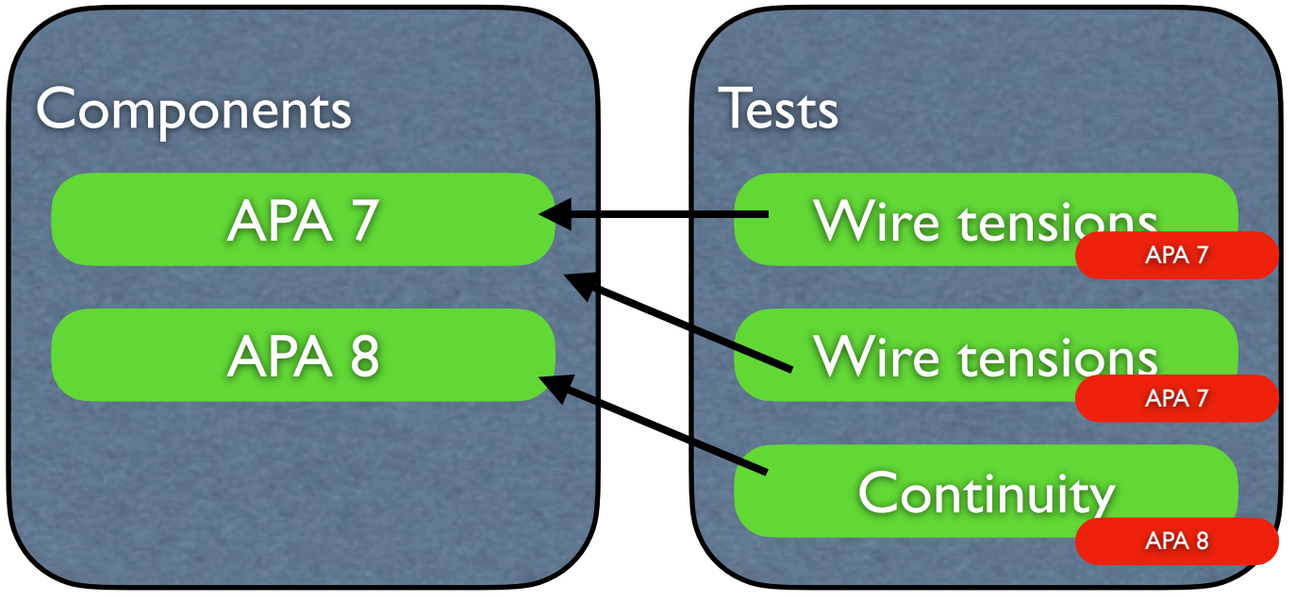
Data entry

Components:
Physical objects

Tests:
Things done to components



Forms describe how data is entered, edited by system experts.

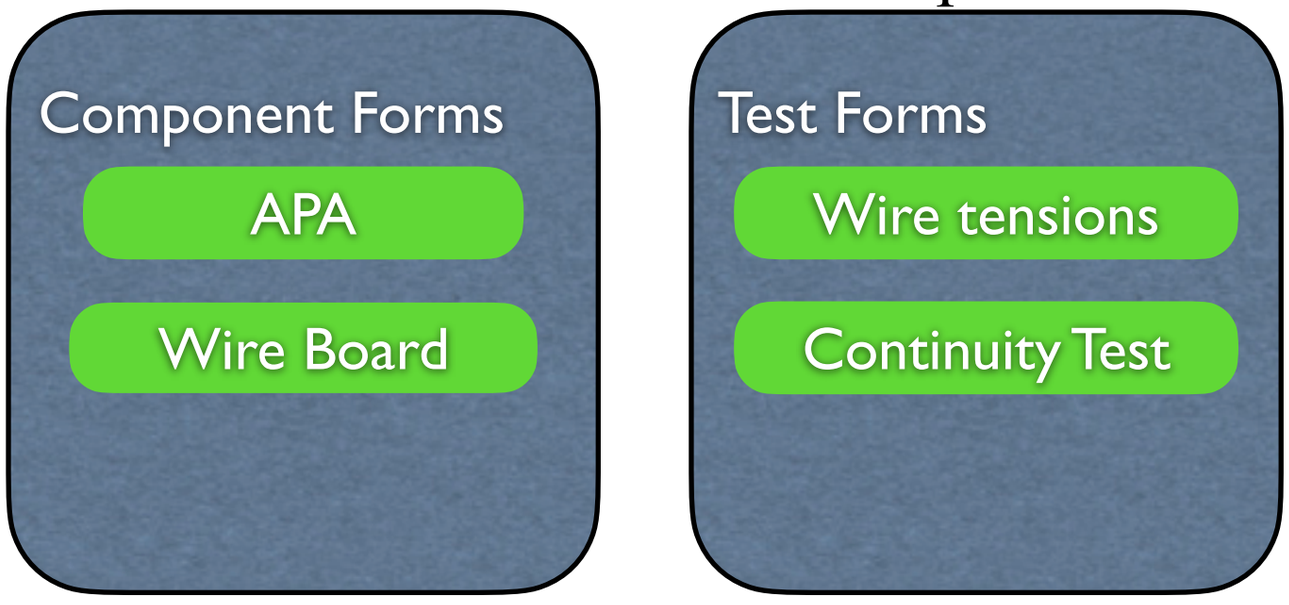


Each test is connected to a component. Each component can be tested more than once.

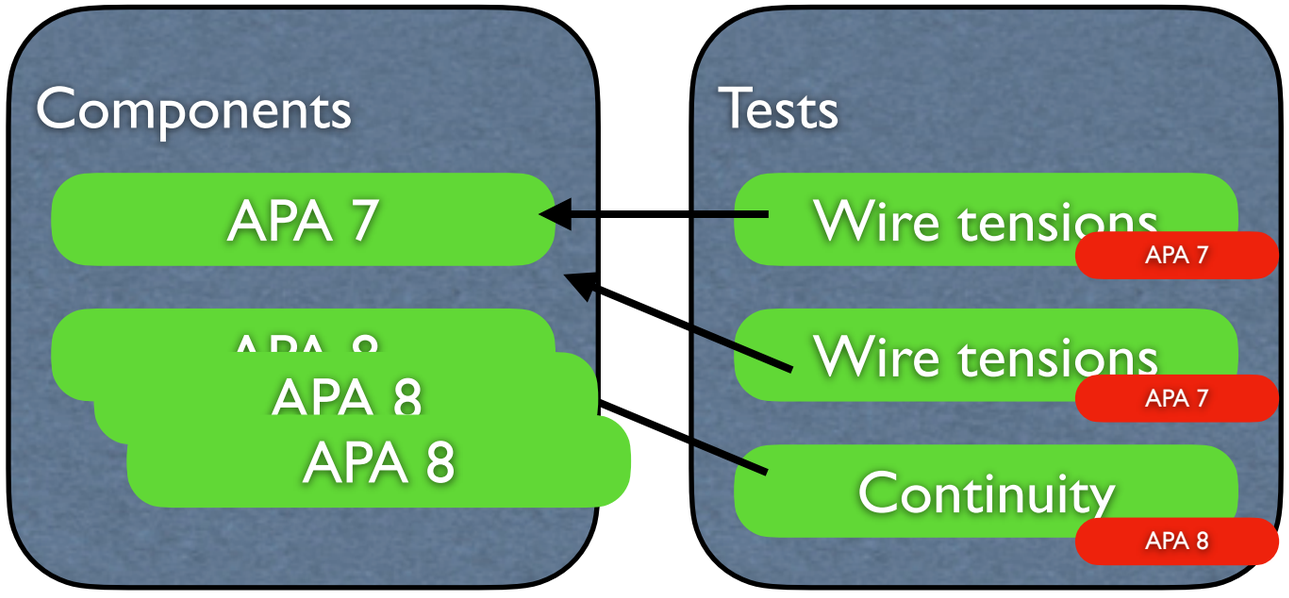
Database "Collections" (tables)

Components:
Physical objects

Tests:
Things done to components



Forms describe how data is entered, edited by system experts.



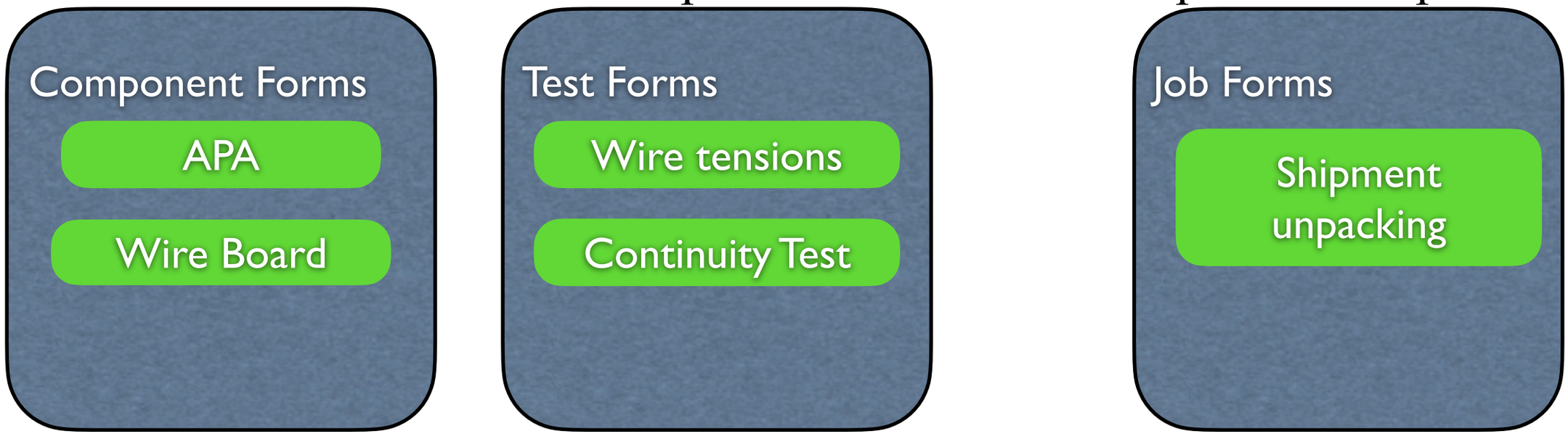
Each test is connected to a component. Each component can be tested more than once.

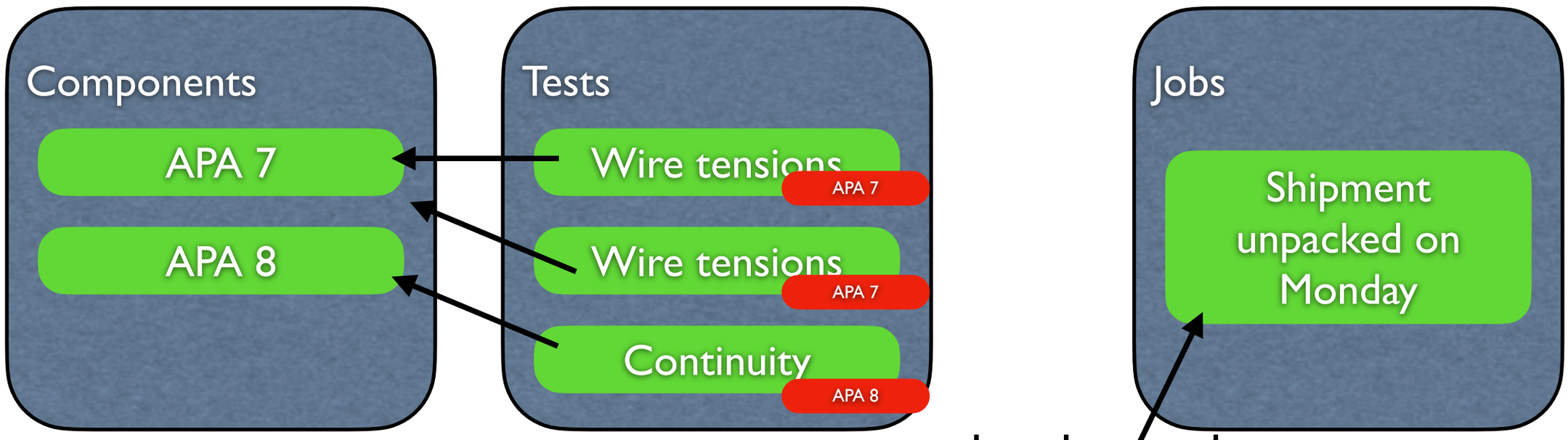
components can evolve, and refer to other (owned or installed) components

Components:
Physical objects

Tests:
Things done to components

Jobs:
Things done, not referenced to specific component





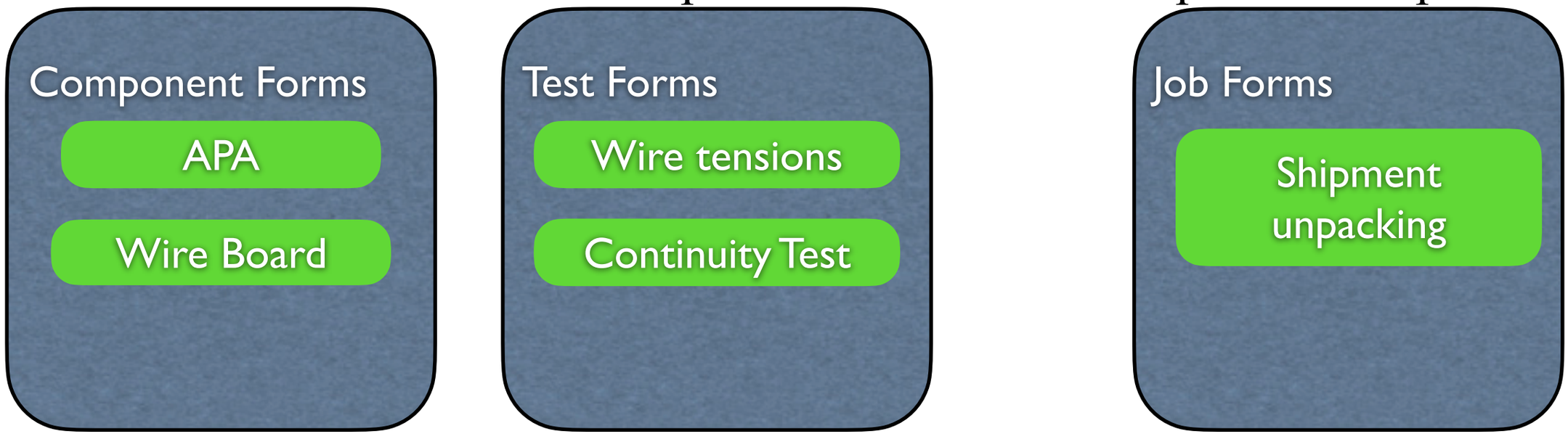
can then be used to create components and tests via user-specified algorithms

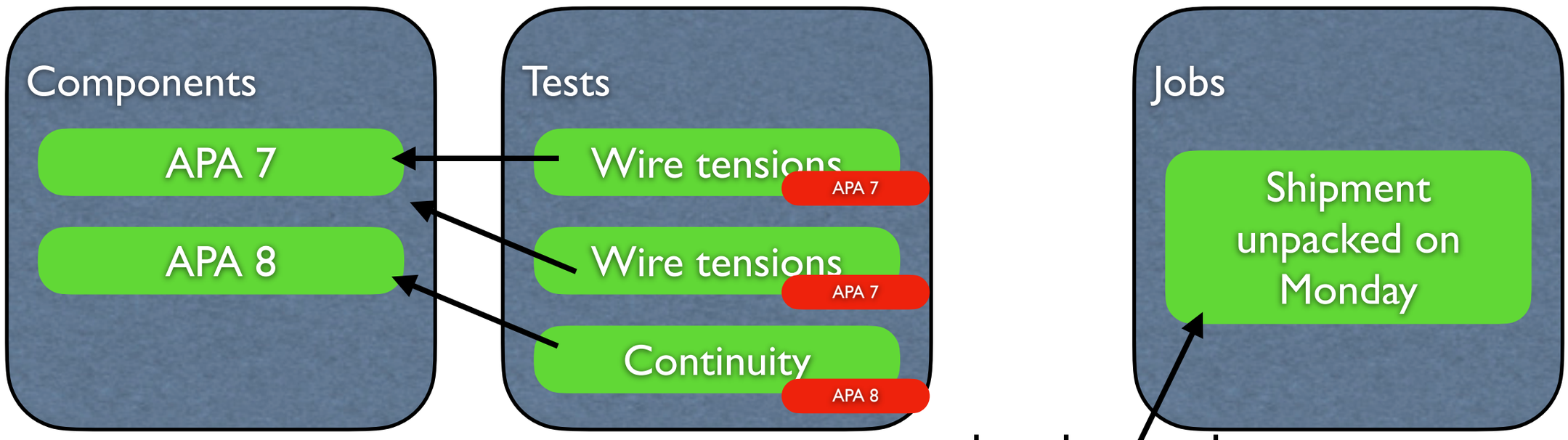
Database “Collections” (tables)

Components:
Physical objects

Tests:
Things done to components

Jobs:
Things done, not referenced to specific component





can then be used to create components and tests via user-specified algorithms

Database “Collections” (tables)

How it would get used

1.Alice logs in

2.Creates a new Widget component in Sietch

3.Enters the Widget serial number 14

4.Prints off the QR code, affixes it to the packagi

5.Bob logs in

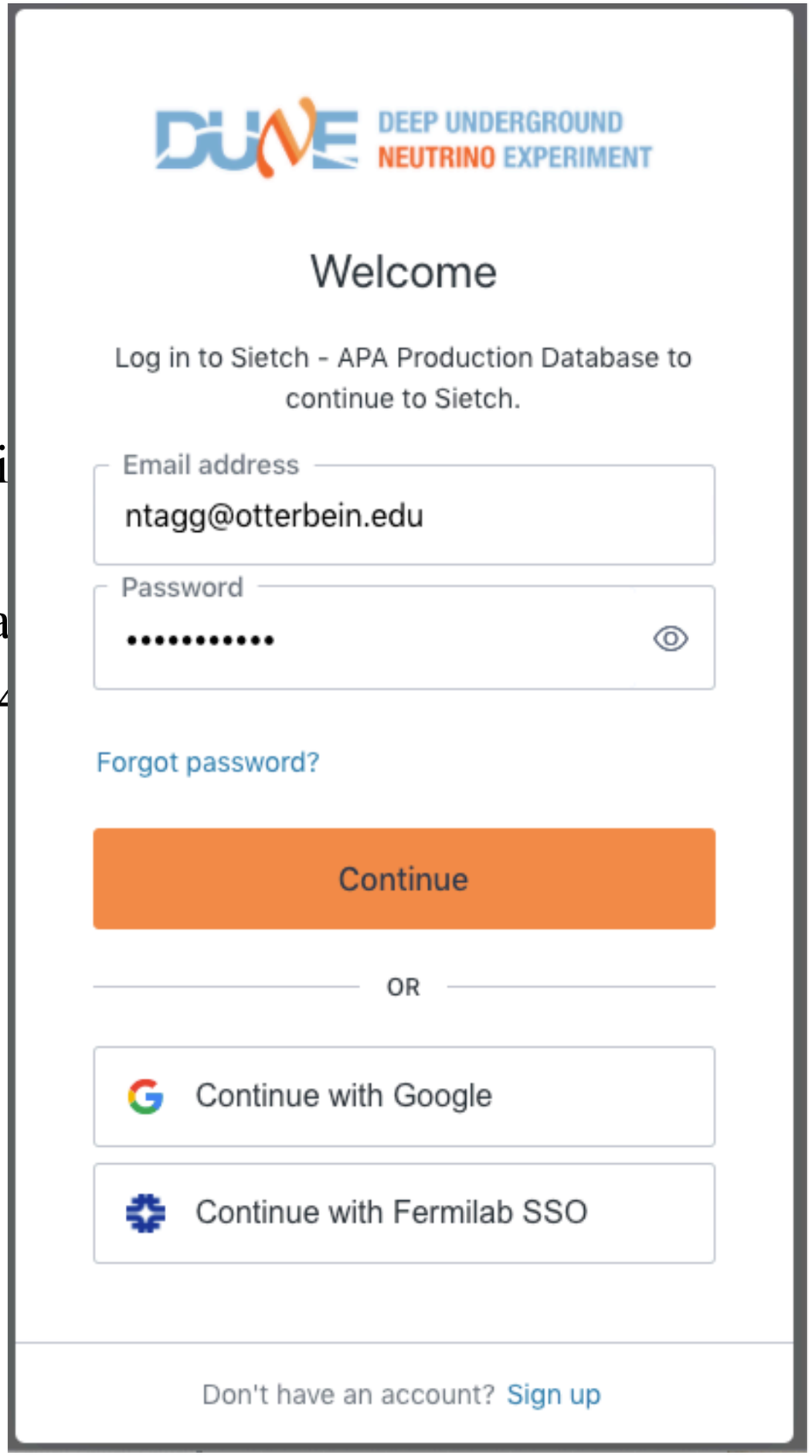
6.Bob scans QR code (or types the first few chara

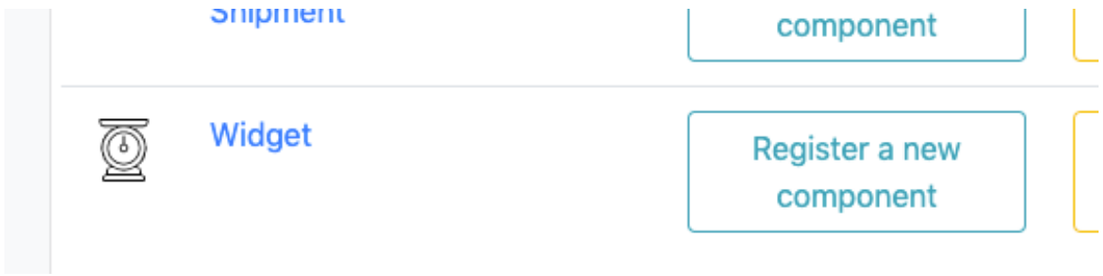
7.Bob starts a ‘Widget integrity test’ for widget 1

8.Bob enters the test results, submits.

9.Widget fails, so Charlie fixes widget

10.Bob redoes test, succeeds





1.Alice logs in

2.Creates a new Widget component in Sietch

3.Enters the Widget serial number 14

4.Prints off the QR code, affixes it to the packaging of the Widget

5.Bob logs in

6.Bob scans QR code (or types the first few characters of the UUID)

7.Bob starts a ‘Widget integrity test’ for widget 14

8.Bob enters the test results, submits.

9.Widget fails, so Charlie fixes widget

10.Bob redoes test, succeeds

How it would get used

1.Alice logs in

2.Creates a new Widget component in Sietch

3.Enters the Widget serial number 14

4.Prints off the QR code, a

5.Bob logs in

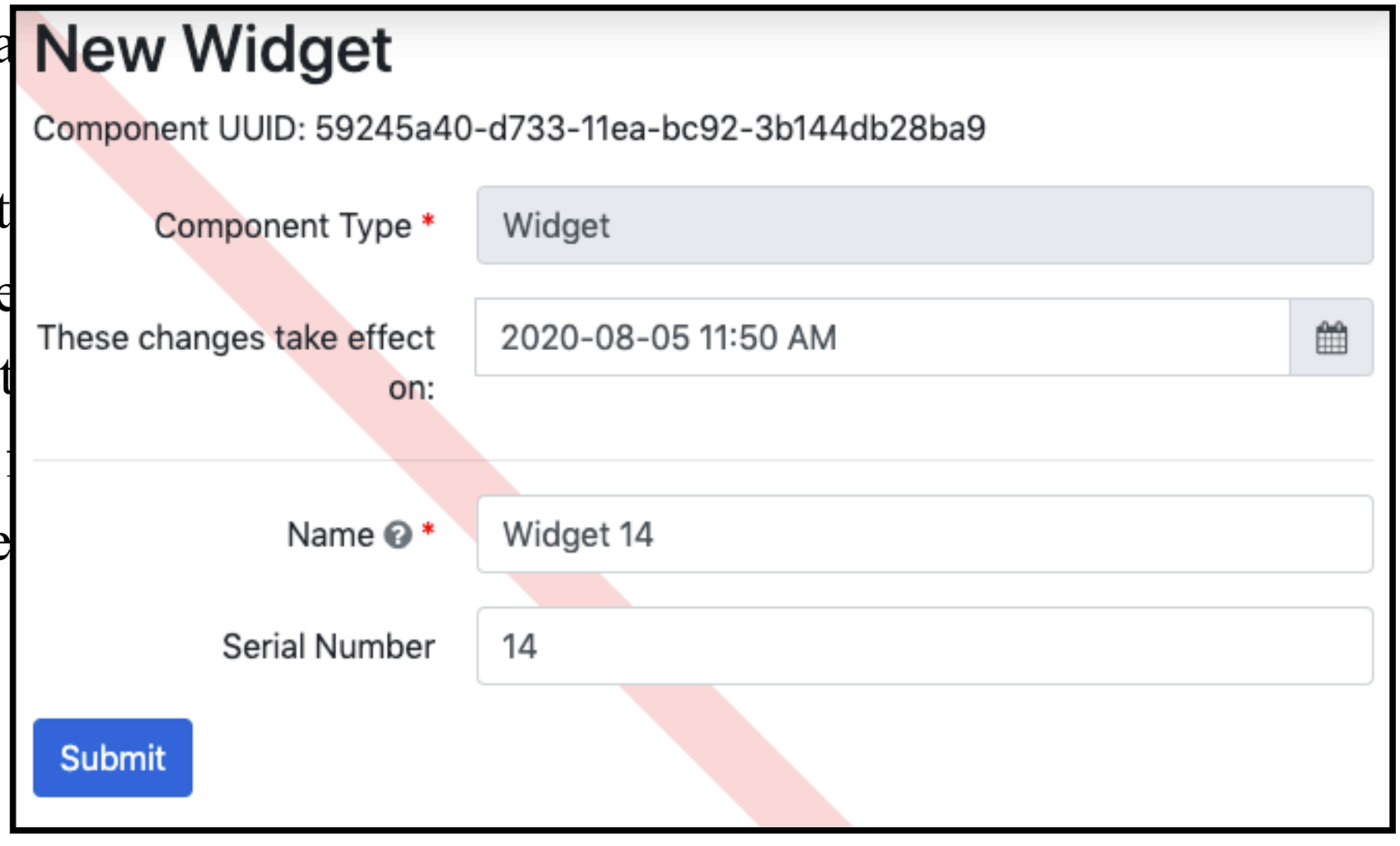
6.Bob scans QR code (or t

7.Bob starts a ‘Widget inte

8.Bob enters the test result

9.Widget fails, so Charlie

10.Bob redoes test, succee



How it would get used

1.Alice logs in

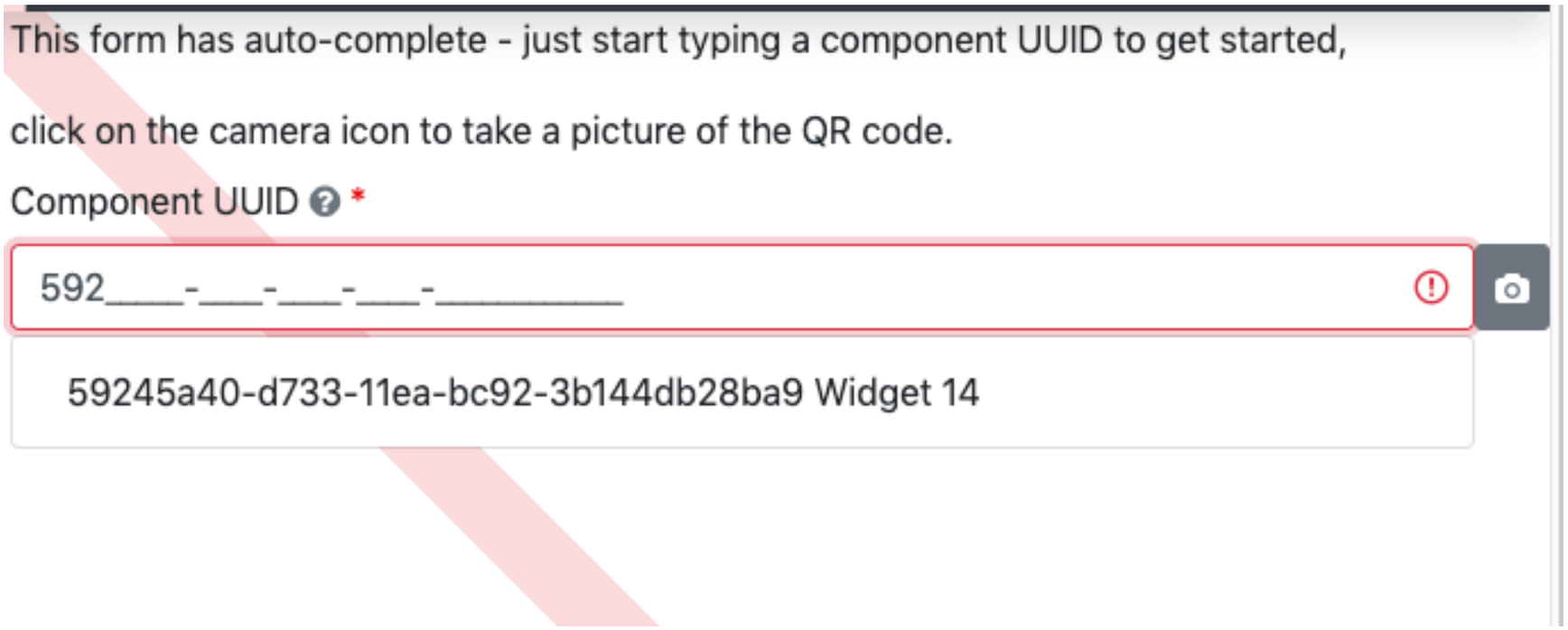
2.Creates a new Widget component in Sietch

3.Enters the Widget serial number 14

4.Prints off the QR code, affixes it to the packaging of the Widget

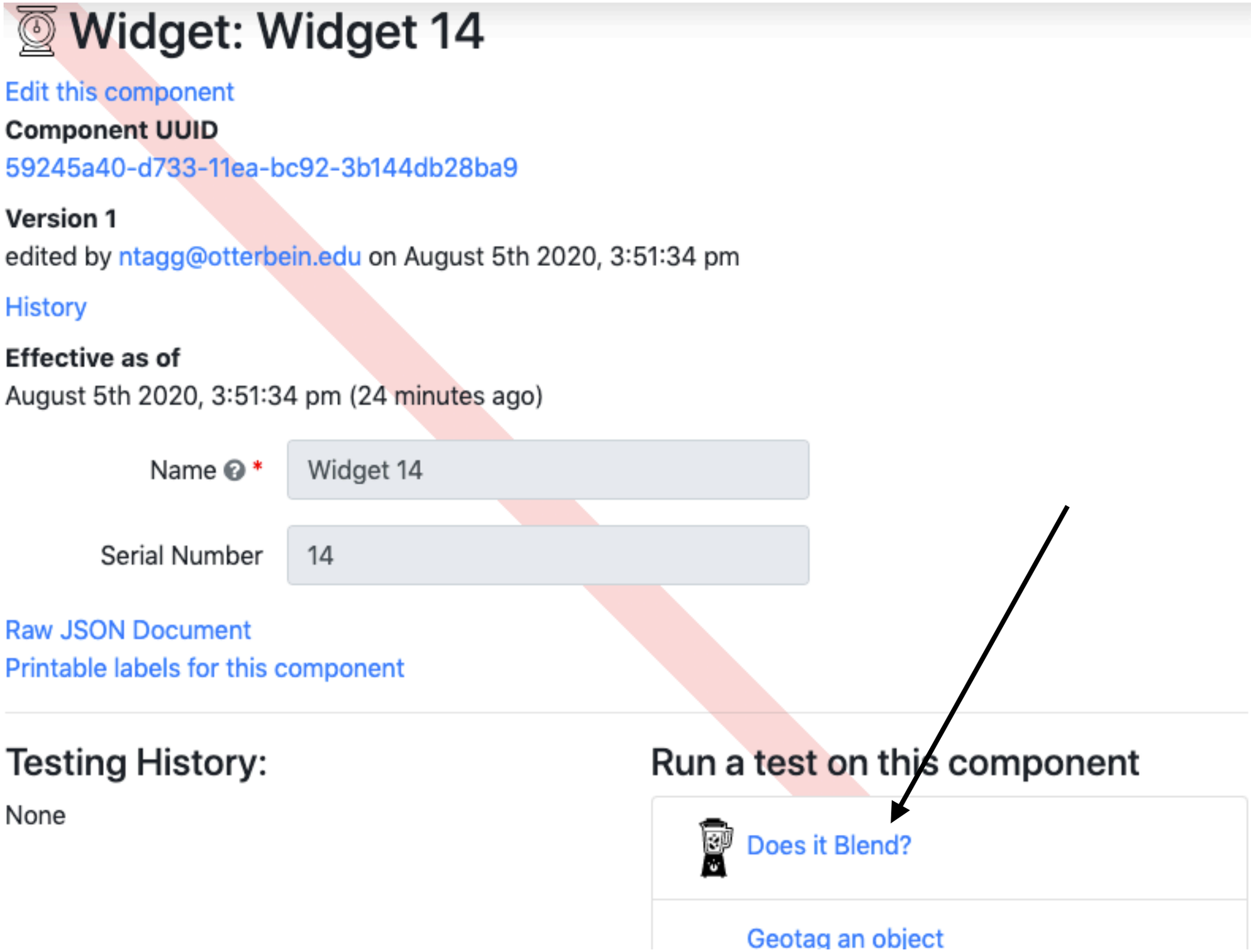
5.Bob logs in

6.Bob scans QR code (or types the first few characters of the UUID)



How it would get used

7.Bob starts a ‘Widget integrity test’ for widget 14



How it would get used

1.Alice logs in

2.Creates a new Widget compo

3.Enters the Widget serial num

4.Prints off the QR code, affixe

5.Bob logs in

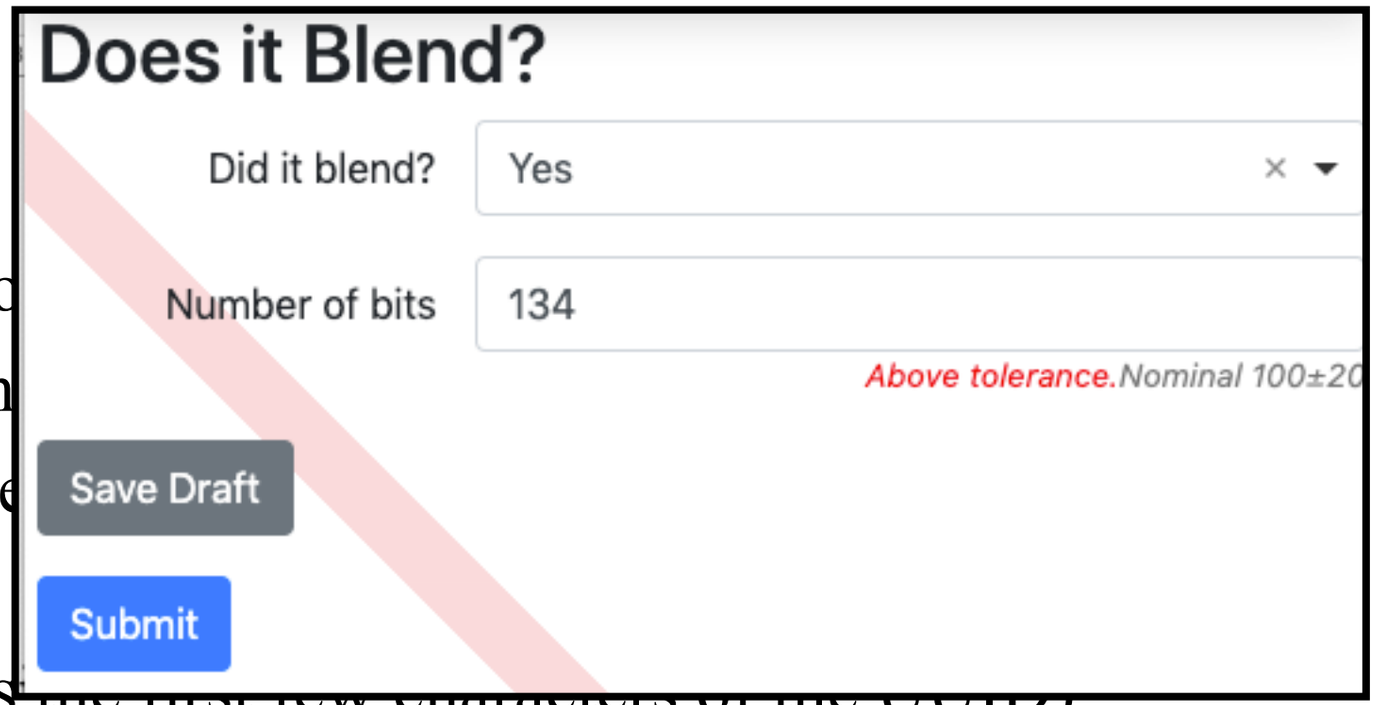
6.Bob scans QR code (or types the first few characters of the UUID)

7.Bob starts a ‘Widget integrity test’ for widget 14

8.Bob enters the test results, submits.

9.Widget fails, so Bob fixes widget

10.Bob redoes test, submits again.



How it would get used

1.Alice logs in
2.Creates a new Widget component in Sietch
3.Enters the Widget serial number 14
4.Prints off the QR code, affixes it to the packaging of the Widget
5.Bob logs in
6.Bob scans QR code (or types the first few characters of the UUID)
7.Bob starts a ‘Widget integrity test’ for widget 14
8.Bob enters the test results, submits.
9.Widget fails, so Bob fixes widget
10.Bob redoes test, submits again.

How it would get used

1.Alice logs in
2.Creates a new Wi
3.Enters the Widget
4.Prints off the QR
5.Bob logs in
6.Bob scans QR cod
7.Bob starts a ‘Widg
8.Bob enters the test
9.Widget fails, so B
10.Bob redoes test,

How it would get used

File storage.
Image storage.
Can take picture with mobile device.

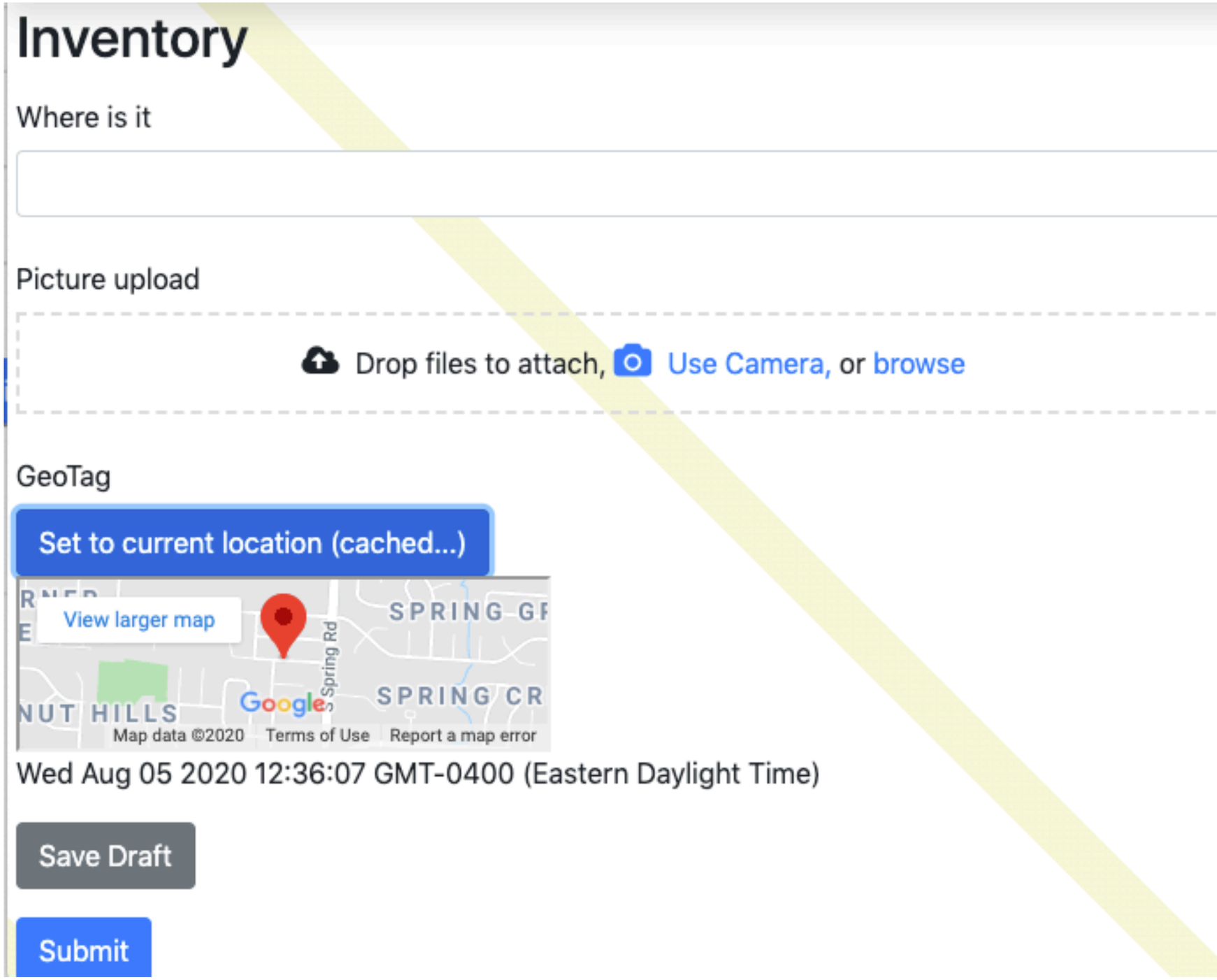
Location tagging

New! Search engine in place

Coming soon: Track component connections

Custom views for array data

Automatic specification warnings



Features

Features

File storage.

Image storage.

Can take picture with mobile device.

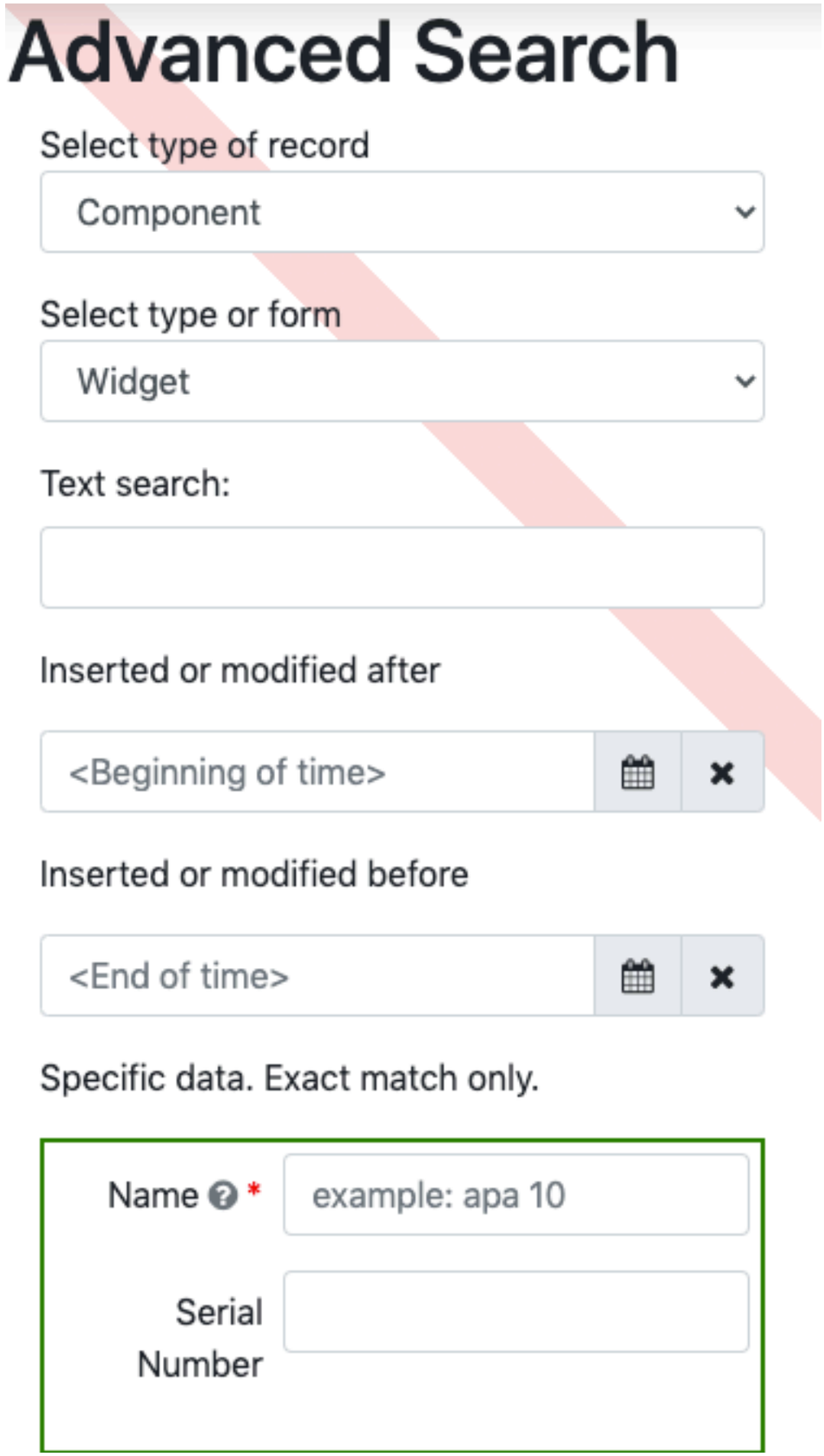
Location tagging

New! Search engine in place

Coming soon: Track component connections

Custom views for array data

Automatic specification warnings



File storage.
Image storage.
Can take picture with mobile device.

Location tagging

New! Search engine in place

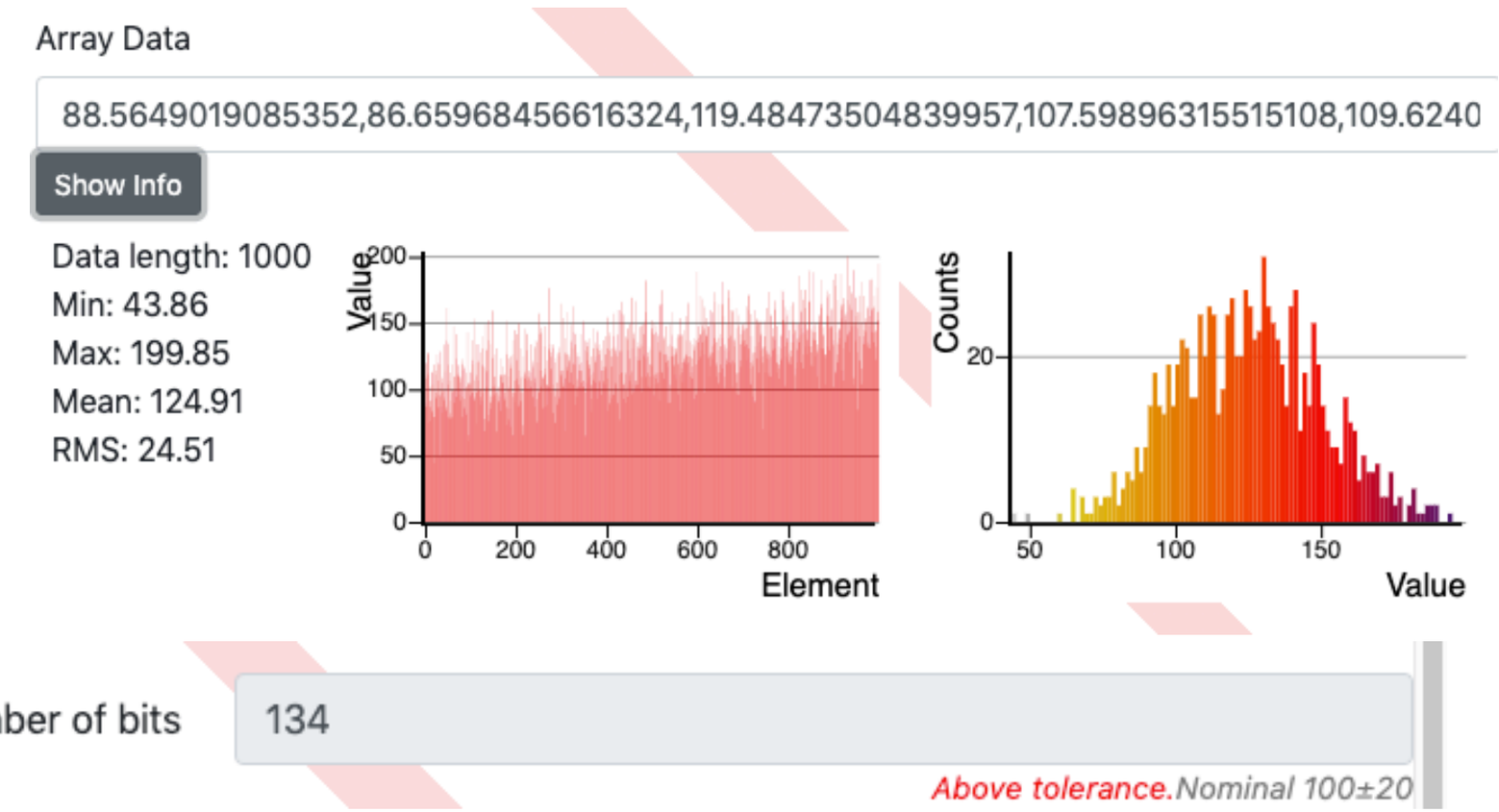
Coming soon: Track component connections

Custom views for array data

Automatic specification warnings

User management system. does NOT require FNAL ID, but can use it.

Documented API for scripts or other applications to upload test results programmatically

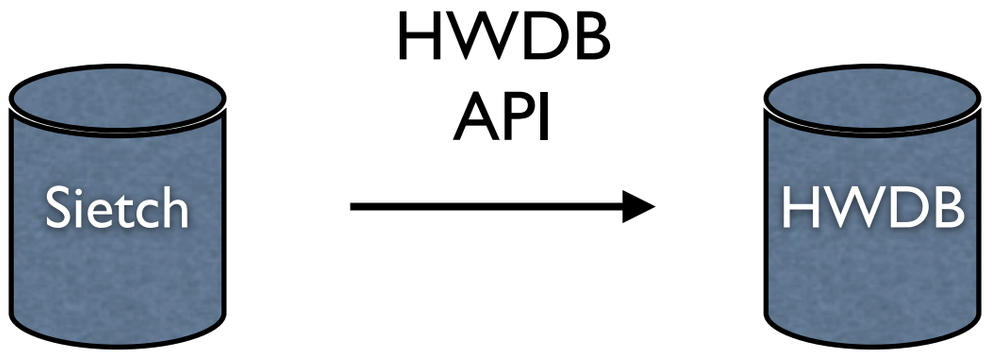


Features

HWDB is the fully sanctioned long-term solution for storing this sort of data.

Sietch uses MongoDB, which Fermilab won't support.

Working model: Sietch exports it's data to HWDB for long-term storage.



mongo postgres

Sietch is ready for real-world testing,
(APA Steel, APA wire boards are moving to this soon)

HWDB is still under construction

Long-term