

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

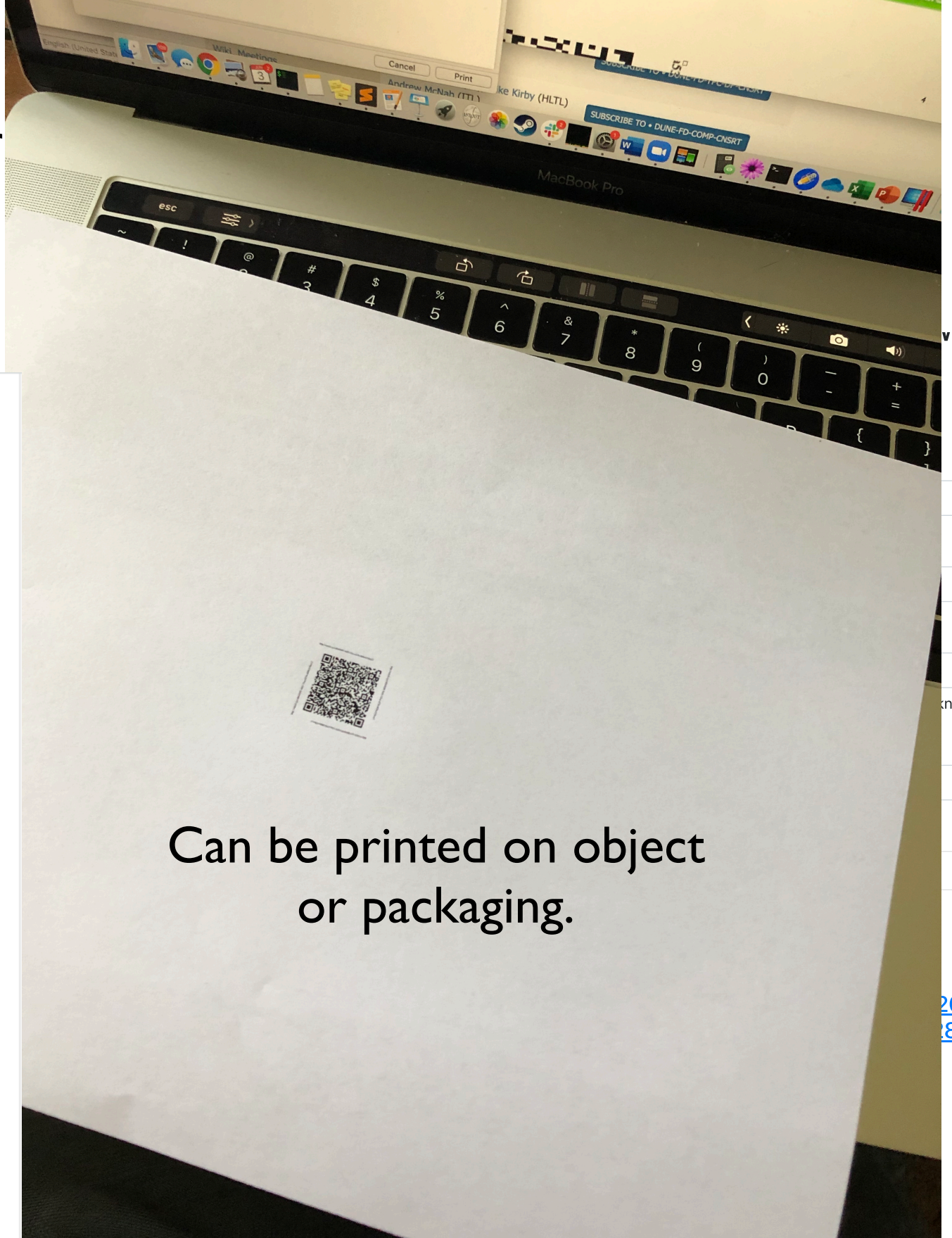
Never delete anything. All objects are either versioned or immutable.

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
A P A I D	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 I 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

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

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	<input type="text"/> Nominal 4.76 ± 0.1	<input type="text"/> Nominal 4.76 ± 0.1
Tongue Thickness	<input type="text"/> Nominal 4.56 ± 0.1 mm	<input type="text"/> Nominal 4.56 ± 0.1
Board Thickness - Tongue Thickness (calculated)	<input type="text"/> min 0.1	<input type="text"/> min 0.1

Step 2: Visual checks

Silkscreen is OK

- ☐ Pass
☐ Fail

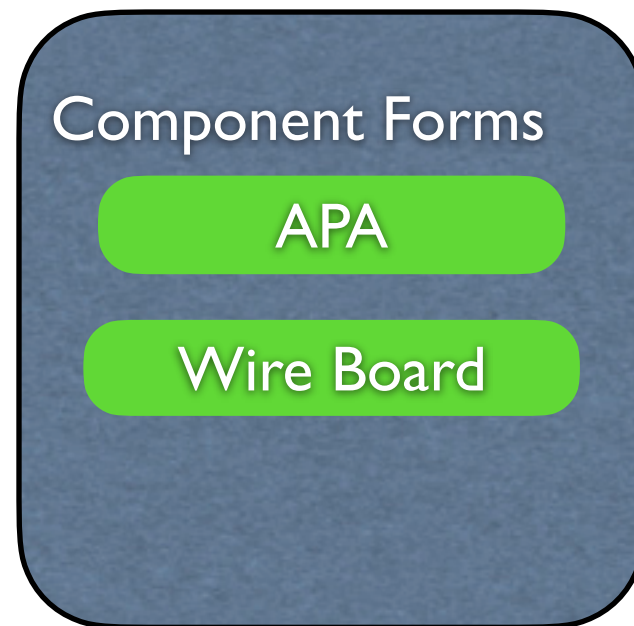
☐ Right Side Cut Corner Vertical ⓘ
Nominal 9.00 mm

☐ Right Side Cut Corner Horizontal ⓘ
Nominal 8.80 mm

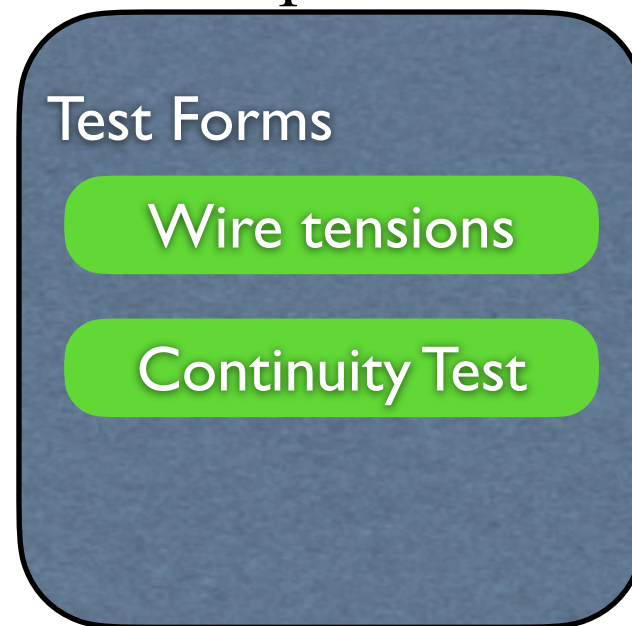
☐ Notch - Vertical ⓘ
Nominal 9.00 mm

☐ Notch - Horizontal ⓘ

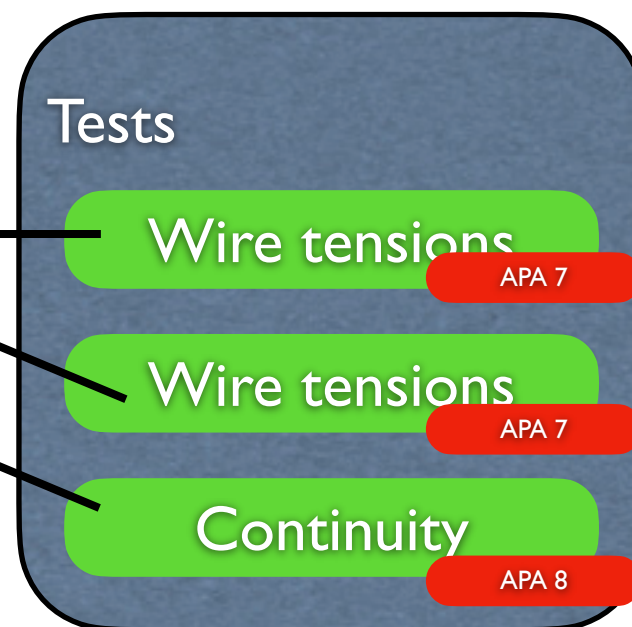
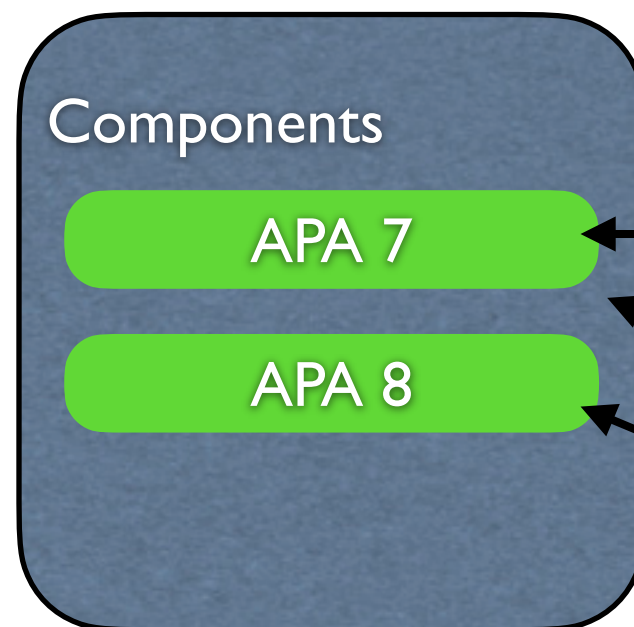
Components:
Physical objects



Tests:
Things done to
components



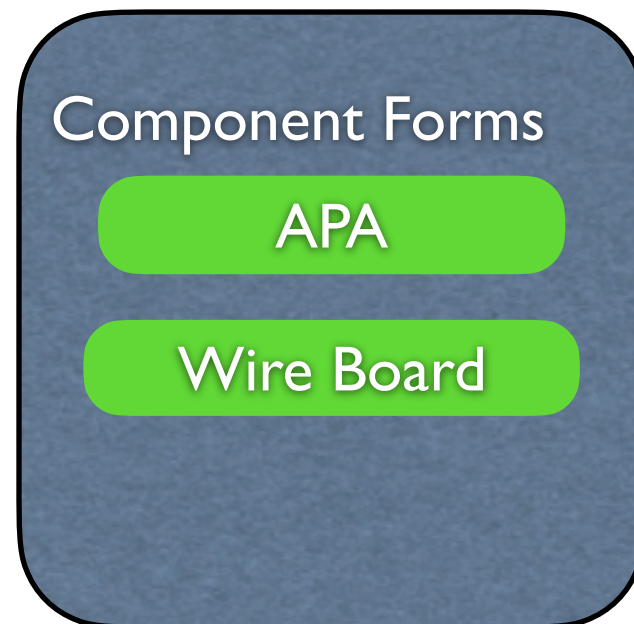
Forms describe how data is entered, edited by system experts.



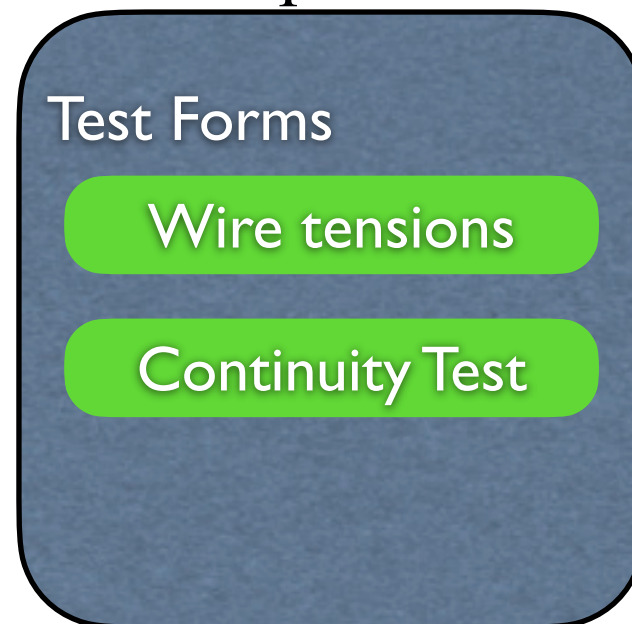
Each test is connected to a component. Each component can be tested more than once.

Database “Collections” (tables)

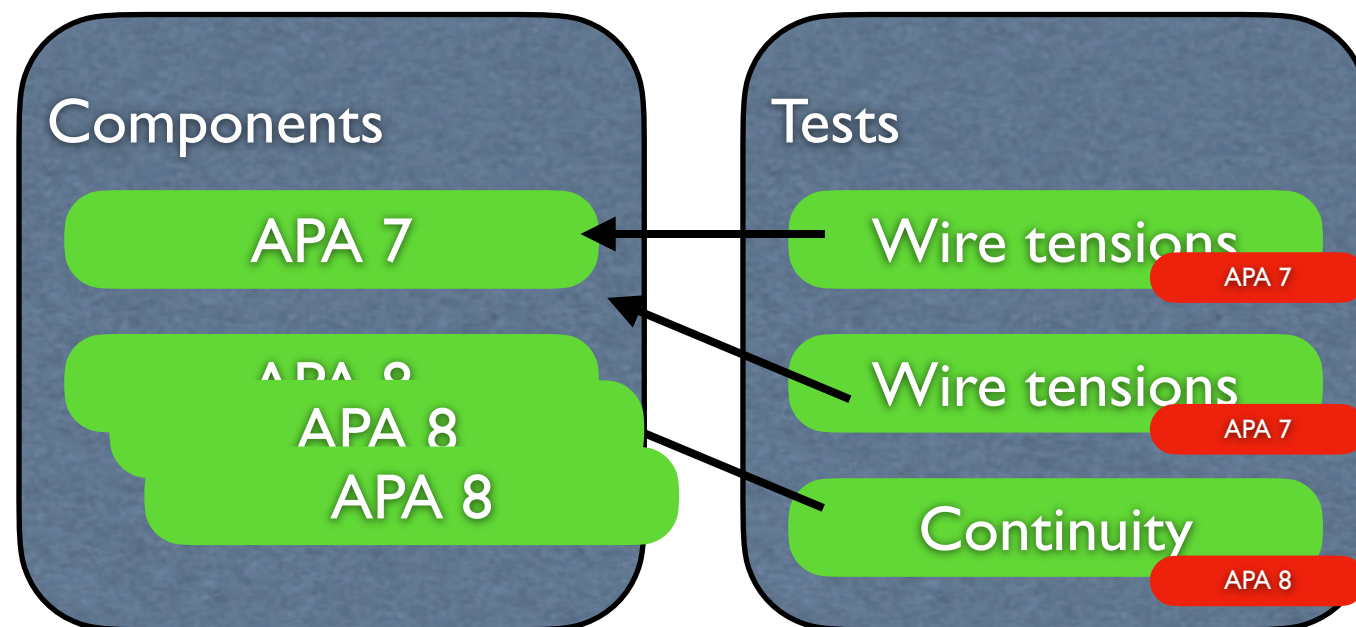
Components:
Physical objects



Tests:
Things done to
components



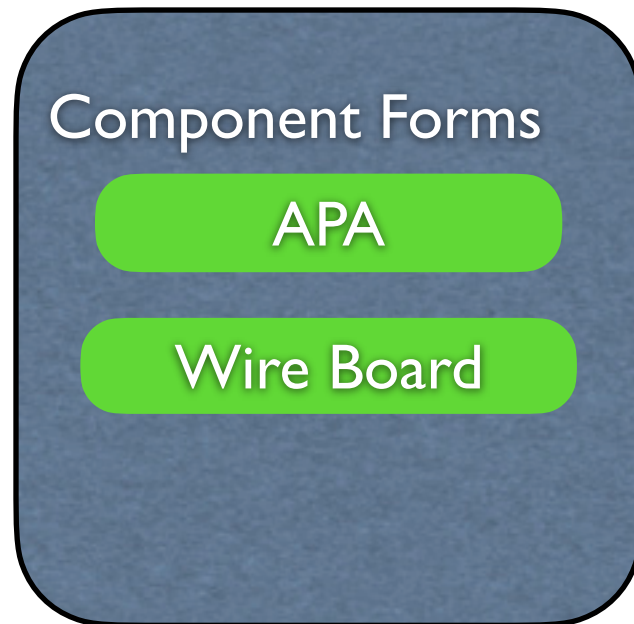
Forms describe how data is entered, edited by system experts.



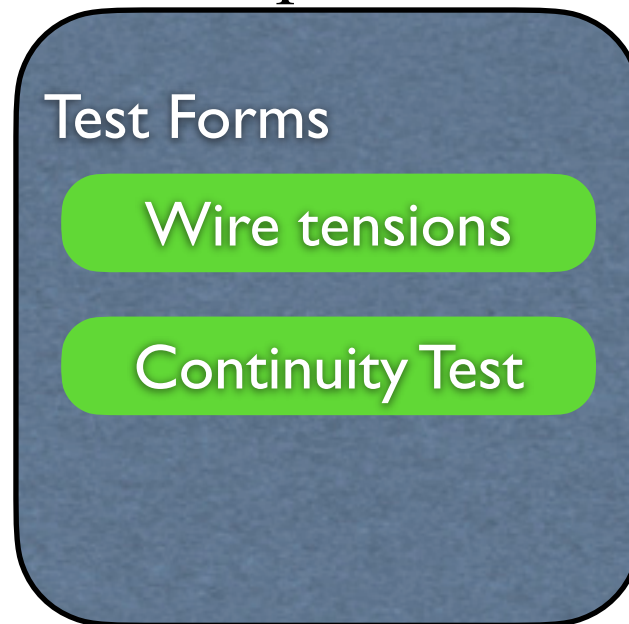
Each test is connected to a component. Each component can be tested more than once.

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

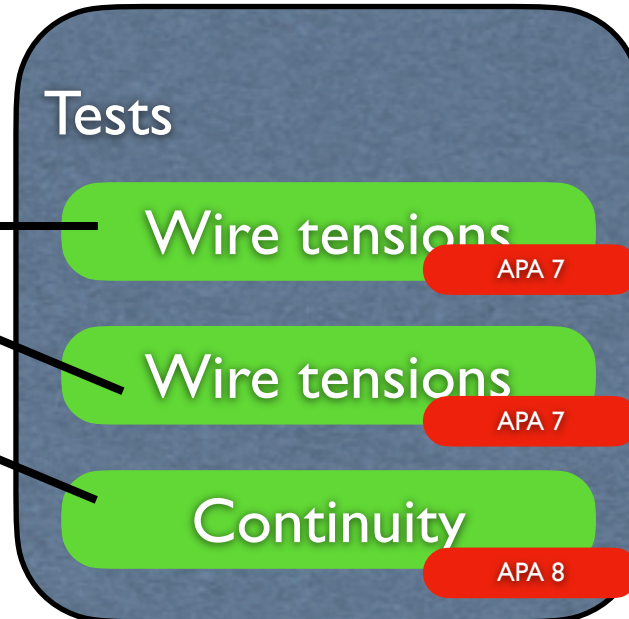
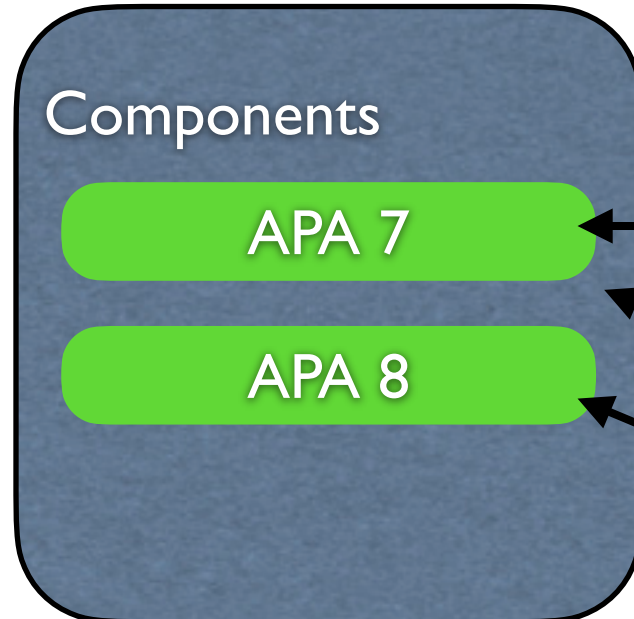
Components:
Physical objects



Tests:
Things done to components



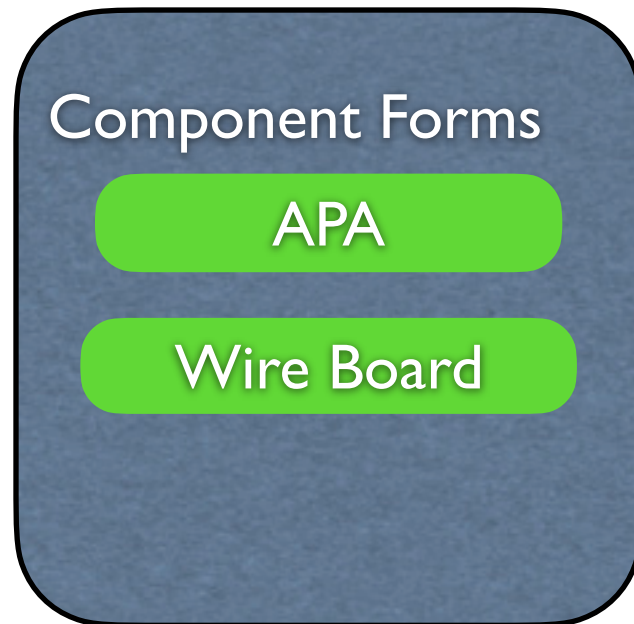
Jobs:
Things done, not referenced to specific component



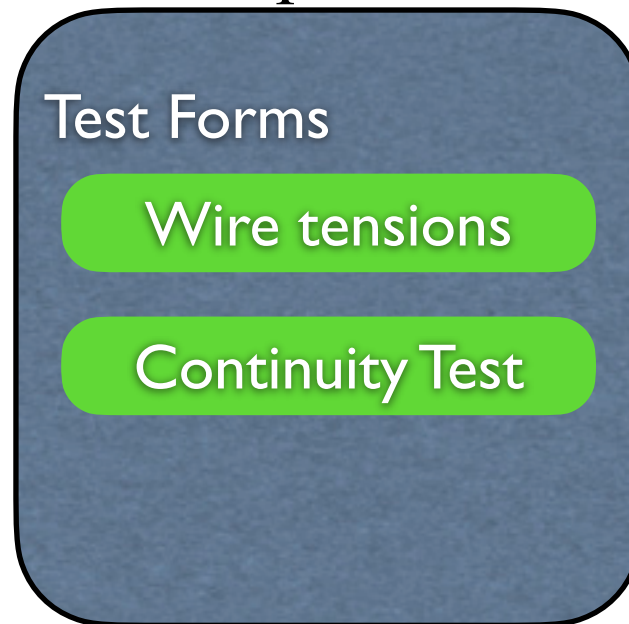
can then be used to create components and tests via user-specified algorithms

Database “Collections” (tables)

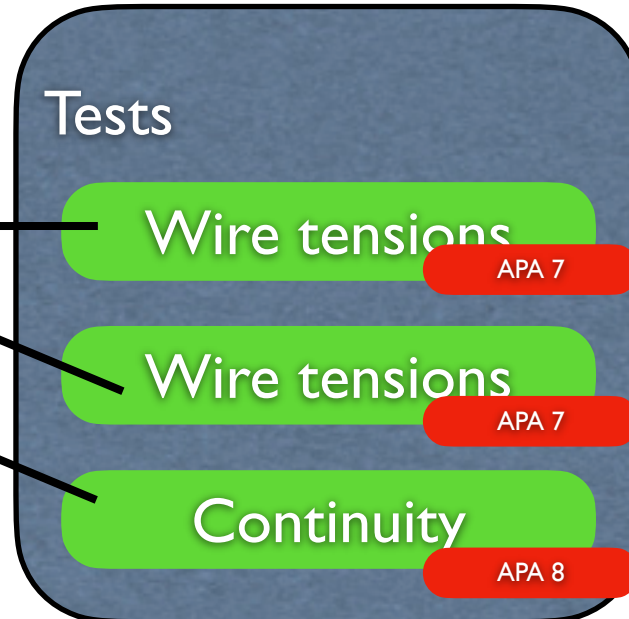
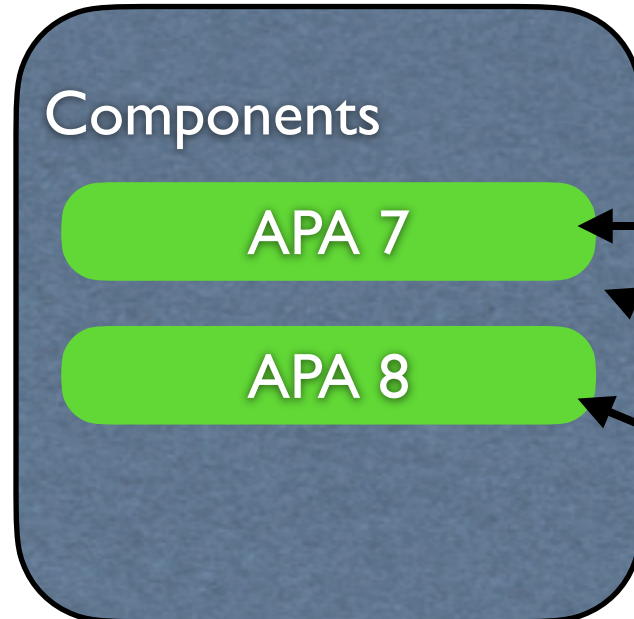
Components:
Physical objects



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)

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

5. Bob logs in

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

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



Welcome

Log in to Sietch - APA Production Database to continue to Sietch.

Email address

ntagg@otterbein.edu

Password

.....



[Forgot password?](#)

Continue

OR



Continue with Google



Continue with Fermilab SSO

Don't have an account? [Sign up](#)

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, and
5. Bob logs in
6. Bob scans QR code (or to
7. Bob starts a 'Widget inte
8. Bob enters the test result
9. Widget fails, so Charlie
10. Bob redoes test, succee

New Widget

Component UUID: 59245a40-d733-11ea-bc92-3b144db28ba9

Component Type * Widget

These changes take effect on: 2020-08-05 11:50 AM

Name ? * Widget 14


Serial Number 14


Submit

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)**

This form has auto-complete - just start typing a component UUID to get started,
click on the camera icon to take a picture of the QR code.


Component UUID  *

592____-____-____-____-_____ 

59245a40-d733-11ea-bc92-3b144db28ba9 Widget 14

How it would get used

7. Bob starts a 'Widget integrity test' for widget 14

 **Widget: Widget 14**


[Edit this component](#)

Component UUID
[59245a40-d733-11ea-bc92-3b144db28ba9](#)

Version 1
edited by [ntag@otterbein.edu](#) on August 5th 2020, 3:51:34 pm

[History](#)

Effective as of
August 5th 2020, 3:51:34 pm (24 minutes ago)

Name  *

Widget 14


Serial Number

14

[Raw JSON Document](#)
[Printable labels for this component](#)

Testing History:
None

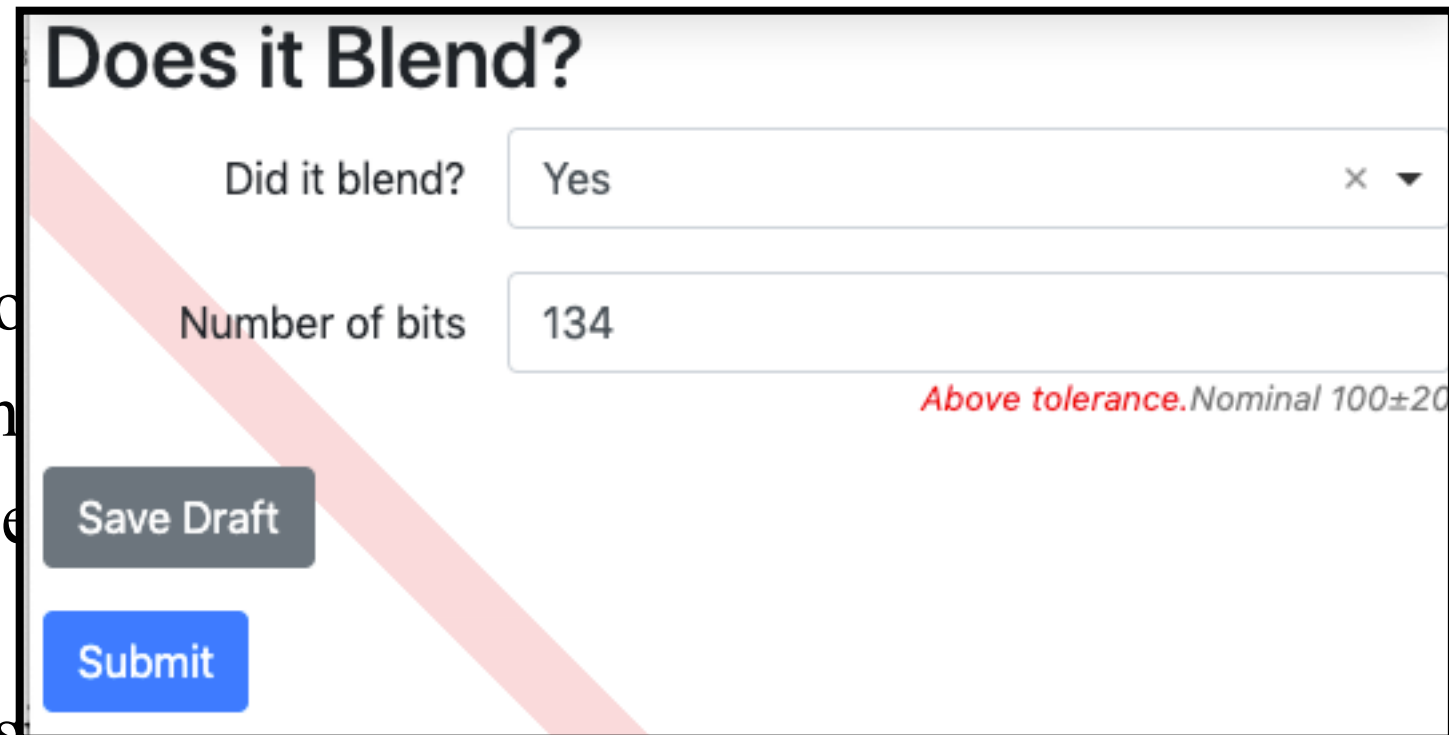
Run a test on this component

 [Does it Blend?](#)

[Geotag an object](#)

How it would get used

1. Alice logs in
2. Creates a new Widget component
3. Enters the Widget serial number
4. Prints off the QR code, affixes it to the widget
5. Bob logs in
6. Bob scans QR code (or types the first few characters of the QR code)
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.




The screenshot shows a web form titled "Does it Blend?". It contains two input fields: "Did it blend?" with the value "Yes" and a close button (x), and "Number of bits" with the value "134". Below these fields is a red text message: "Above tolerance. Nominal 100±20". At the bottom of the form are two buttons: "Save Draft" (grey) and "Submit" (blue). A thick red diagonal line is drawn across the entire form area.

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 Widget
3. Enters the Widget details
4. Prints off the QR code
5. Bob logs in
6. Bob scans QR code
7. Bob starts a 'Widget test'
8. Bob enters the test results
9. Widget fails, so Bob retests
10. Bob redoes test, succeeds

 **Widget: Widget 14**


[Edit this component](#)

Component UUID
[59245a40-d733-11ea-bc92-3b144db28ba9](#)

Version 1
edited by [ntagg@otterbein.edu](#) on August 5th 2020, 3:51:34 pm

[History](#)

Effective as of
August 5th 2020, 3:51:34 pm (40 minutes ago)

Name  *

Widget 14

Serial Number

14


[Raw JSON Document](#)

[Printable labels for this component](#)

Testing History:

- [Does it Blend?](#) (on Aug 05 2020 by [ntagg@otterbein.edu](#))
- [Does it Blend?](#) (on Aug 05 2020 by [ntagg@otterbein.edu](#))

Run a test on this component

 [Does it Blend?](#)

[Prototype APA Gwind](#)

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

Inventory

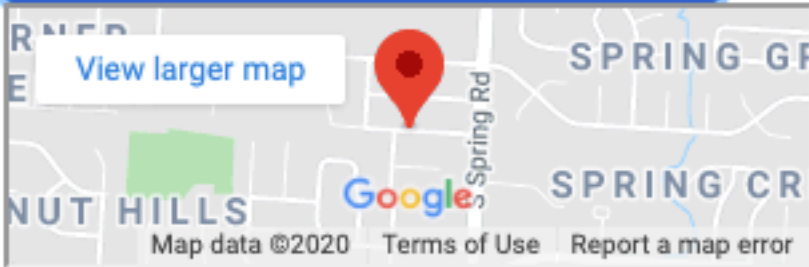
Where is it

Picture upload

 Drop files to attach,  Use Camera, or [browse](#)

GeoTag

[Set to current location \(cached...\)](#)

[View larger map](#)

Wed Aug 05 2020 12:36:07 GMT-0400 (Eastern Daylight Time)

[Save Draft](#)

[Submit](#)

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

Advanced Search

Select type of record

Component

Select type or form

Widget

Text search:

Inserted or modified after

<Beginning of time>

Inserted or modified before

<End of time>

Specific data. Exact match only.

Name ? * example: apa 10

Serial Number

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.

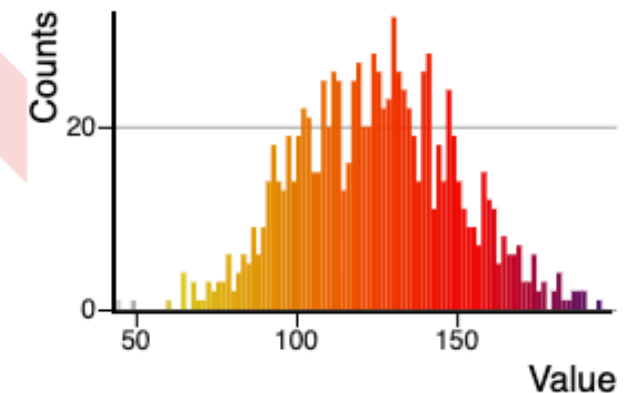
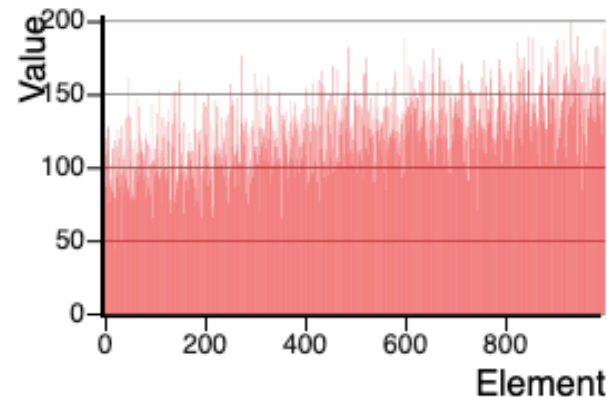
Documented API for scripts or other
applications to upload test results
programmatically

Array Data

88.5649019085352,86.65968456616324,119.48473504839957,107.59896315515108,109.6240

Show Info

Data length: 1000
Min: 43.86
Max: 199.85
Mean: 124.91
RMS: 24.51



Number of bits

134

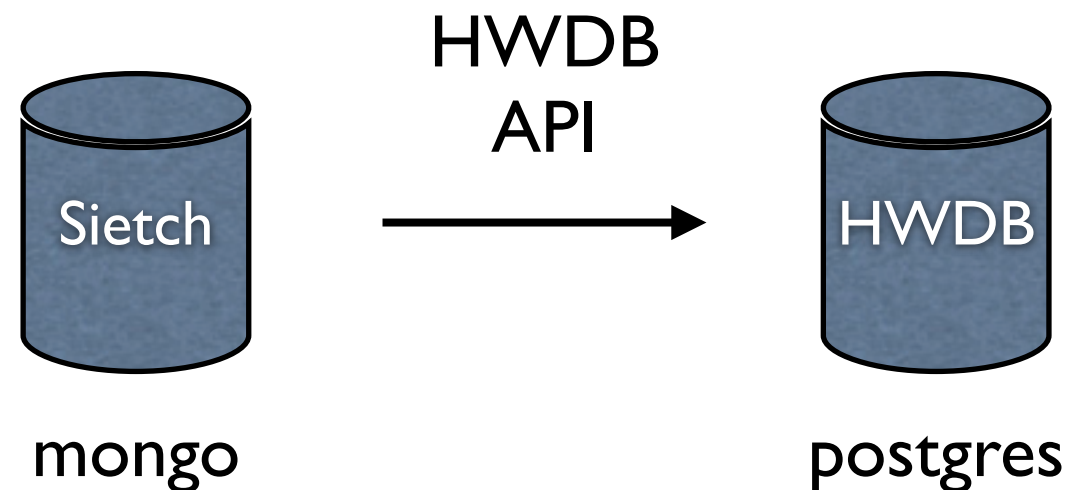
Above tolerance. Nominal 100 ± 20

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.



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

HWDB is still under construction

Long-term