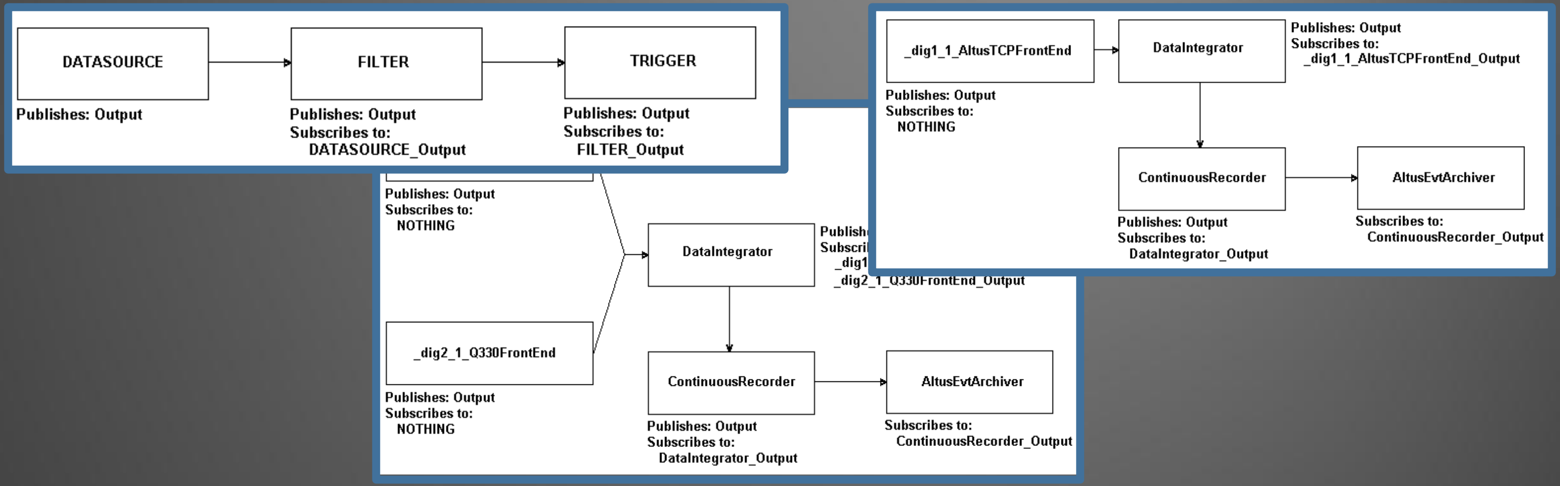
Rockhound
Concept and Data Flow

Antelope/Kinemetrics User's Group
5/29-6/1, 2017



DATASOURCE

Publishes: Output

FILTER

Publishes: Output
Subscribes to:
DATASOURCE_Output

TRIGGER

Publishes: Output
Subscribes to:
FILTER_Output

Publishes: Output
Subscribes to:
NOTHING

DataIntegrator

Publish
Subscri
_dig1
_dig2_1_Q330FrontEnd_Output

_dig2_1_Q330FrontEnd

Publishes: Output
Subscribes to:
NOTHING

ContinuousRecorder

Publishes: Output
Subscribes to:
DataIntegrator_Output

AltusEvtArchiver

Subscribes to:
ContinuousRecorder_Output

_dig1_1_AltusTCPFrontEnd

Publishes: Output
Subscribes to:
NOTHING

DataIntegrator

Publishes: Output
Subscribes to:
_dig1_1_AltusTCPFrontEnd_Output

ContinuousRecorder

Publishes: Output
Subscribes to:
DataIntegrator_Output

AltusEvtArchiver

Subscribes to:
ContinuousRecorder_Output

Dennis Pumphrey

Manager, Software Engineering

Kinemetrics, Inc.

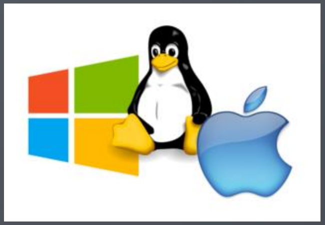
So What is Rockhound$_1$

Rockhound is a platform-independent, digitizer-neutral and format-neutral data collection and processing software used across a wide variety of products

So What is Rockhound$_2$

Where is Rockhound used?

- PC-based data collection systems (OASIS)
  - Includes Windows, Linux, MAC
- Slate-based data collection systems
  - e.g.: Collecting data from a Q330 and generating SEEDLink
- Digitizers: Basalt, Granite, Obsidian, Etna2
  - As the on-board processing, recording, and telemetry software



So What is Rockhound$_3$

Key Points

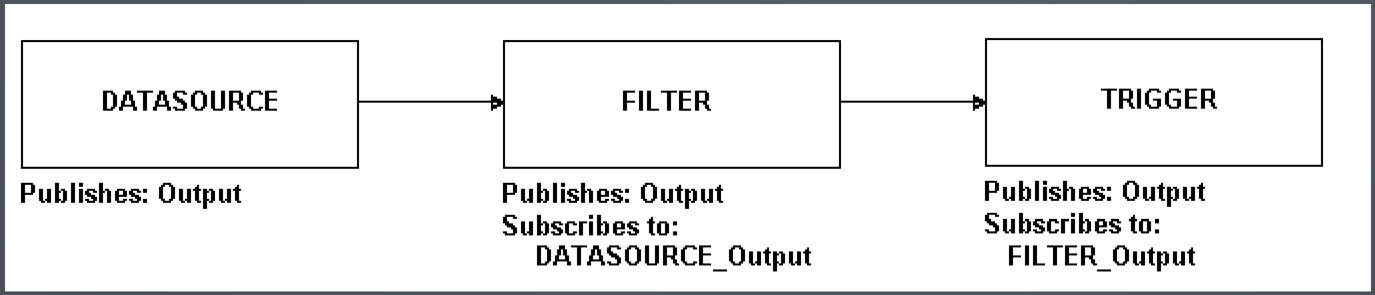
- Highly configurable
- Written in Java to be platform independent
- License included for Kinemetrics hardware
- User extensible

- Configurable via web, RockTalk, or ASCII config
- Consists of Rockhound runtime, and companion utilities like RockTalk, RockMonitor

So What is Rockhound$_4$

How Does it Work?

Rockhound operates by chaining together groups of “modules” into a “layout”. Modules exchange data using a subscription/publication model and pass format neutral data packets:



DATASOURCE

Publishes: Output

FILTER

Publishes: Output
Subscribes to:
DATASOURCE_Output

TRIGGER

Publishes: Output
Subscribes to:
FILTER_Output

So What is Rockhound$_5$

Module Types

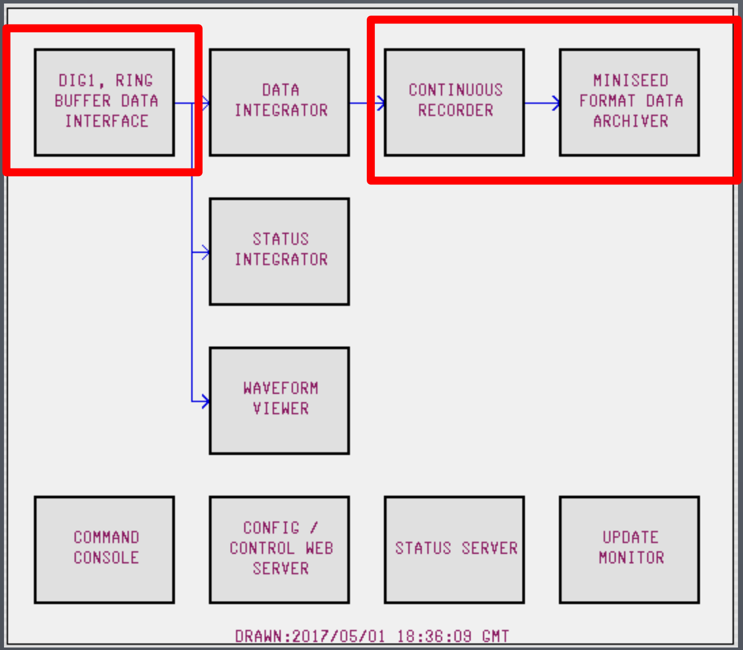
Modules are one of four basic types:

- Data Sources (Data acquired from a Q330)
- Data Processors (voters, filters)
- Data Endpoints (Data recording or Telemetry)
- Modules not in the data flow (Such as Web Server)

Modules should have one well-defined “job”

Examples$_1$

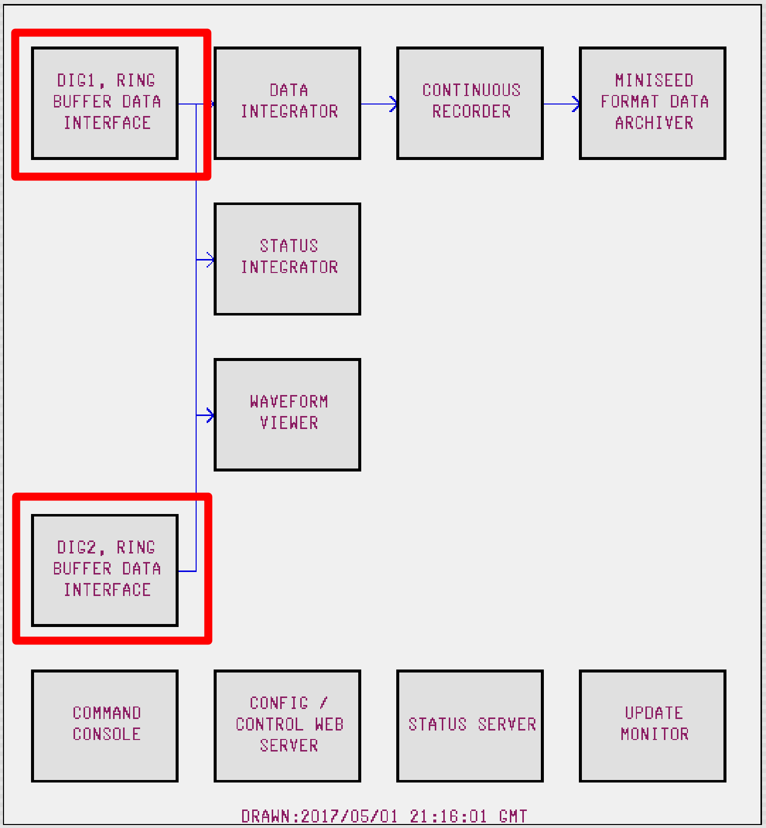
Single Source Continuous Recording From a Ring Buffer



DRAWN:2017/05/01 18:36:09 GMT

Examples$_2$

Continuous Recording From Two Ring Buffers



DIG1, RING BUFFER DATA INTERFACE

DATA INTEGRATOR

CONTINUOUS RECORDER

MINISEED FORMAT DATA ARCHIVER

STATUS INTEGRATOR

WAVEFORM VIEWER

DIG2, RING BUFFER DATA INTERFACE

COMMAND CONSOLE

CONFIG / CONTROL WEB SERVER

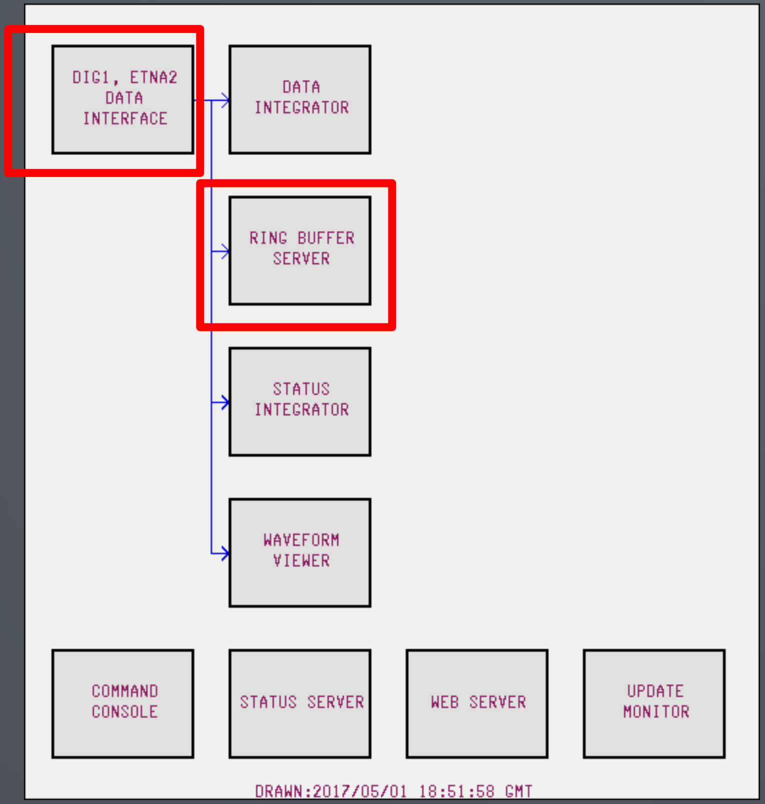
STATUS SERVER

UPDATE MONITOR

DRAWN:2017/05/01 21:16:01 GMT

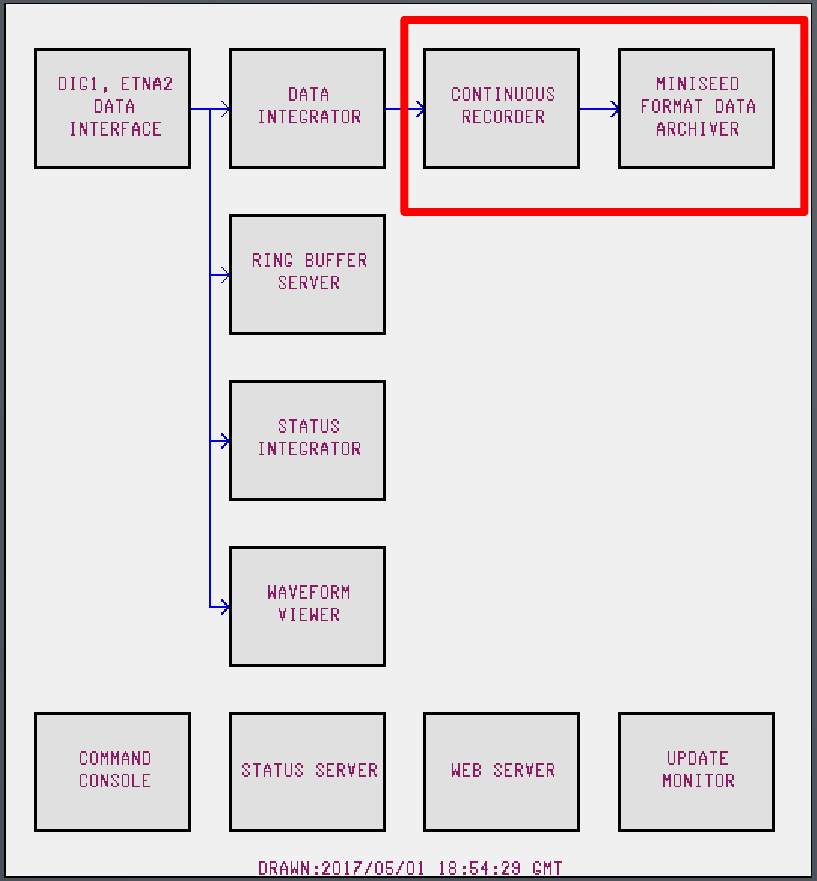
Examples$_3$

Etna2 Telemetry Only



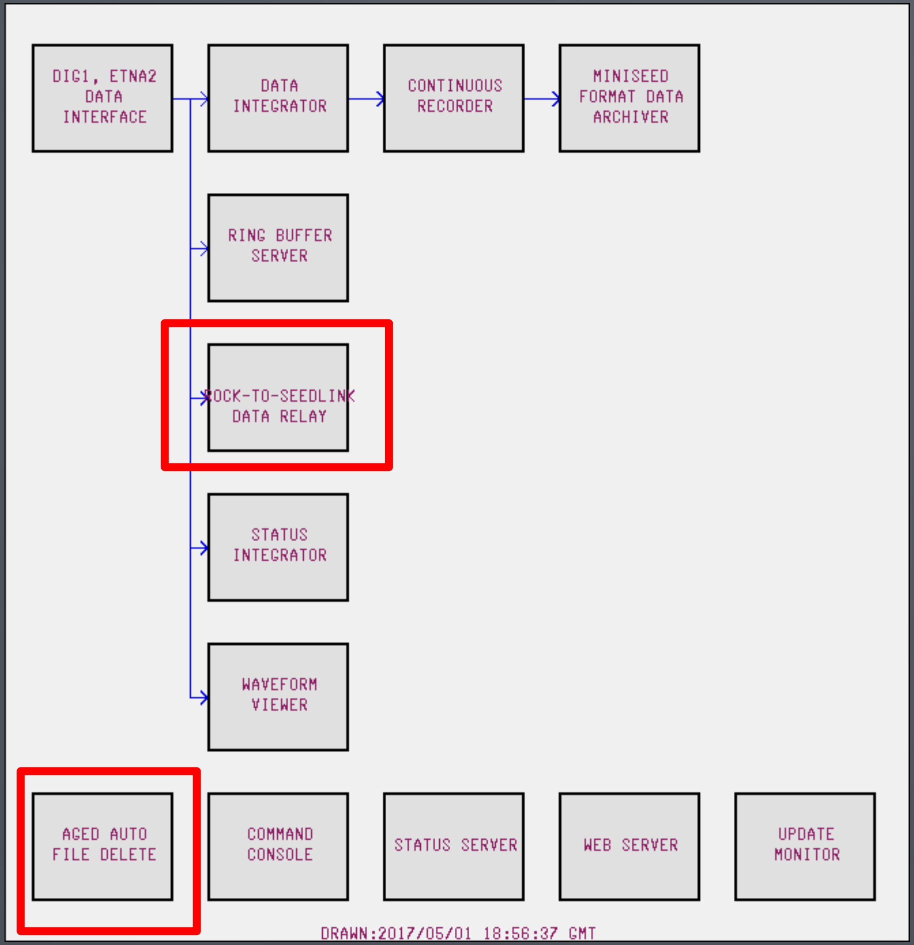
Examples$_4$

Etna2 Telemetry and Continuous Recording



Examples$_5$

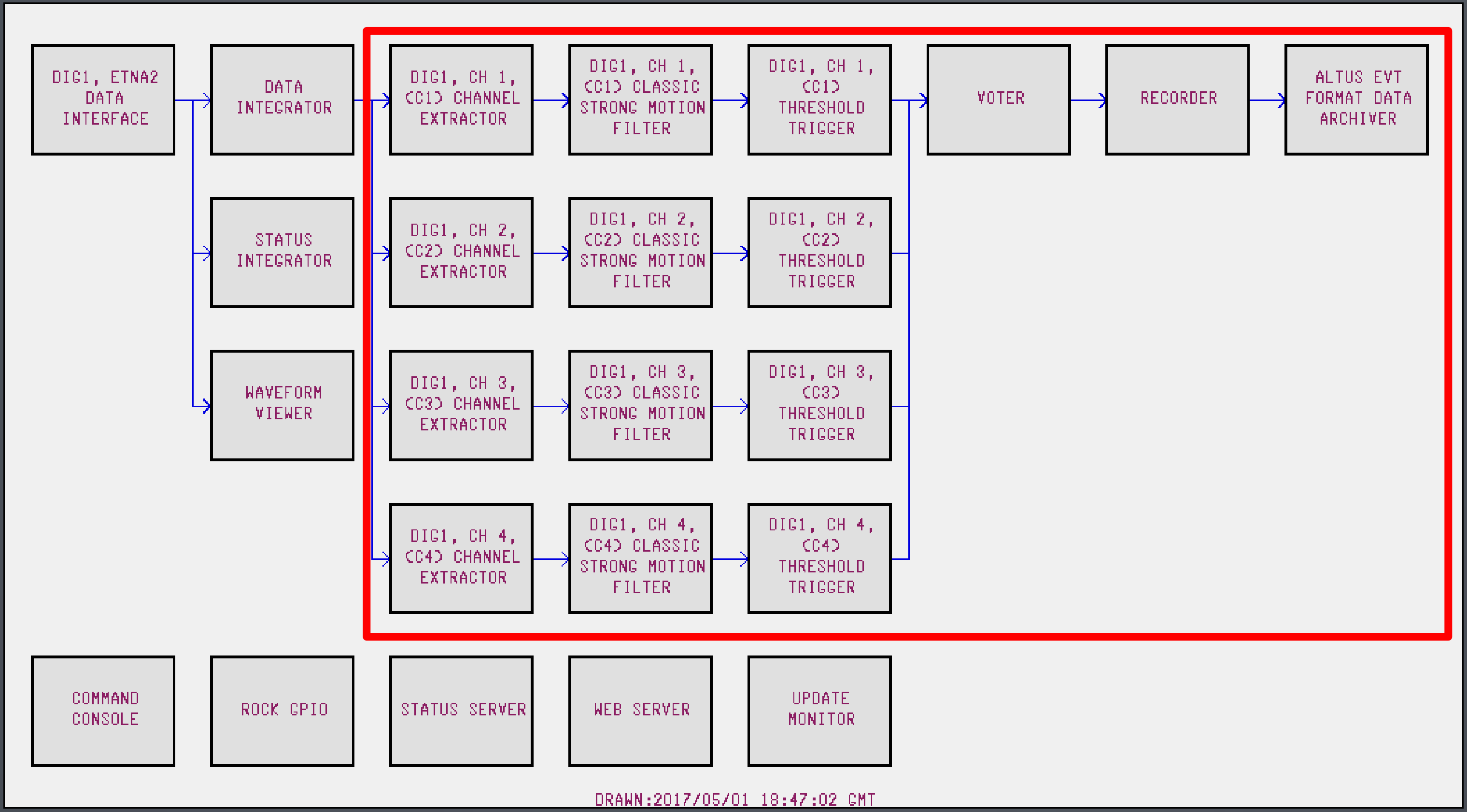
Adding SEEDLink and Auto-File Delete



DRAWN:2017/05/01 18:56:37 GMT

Examples$_6$

Event Recording Etna2



DIG1, ETNA2 DATA INTERFACE

DATA INTEGRATOR

STATUS INTEGRATOR

WAVEFORM VIEWER

DIG1, CH 1, (C1) CHANNEL EXTRACTOR

DIG1, CH 1, (C1) CLASSIC STRONG MOTION FILTER

DIG1, CH 1, (C1) THRESHOLD TRIGGER

DIG1, CH 2, (C2) CHANNEL EXTRACTOR

DIG1, CH 2, (C2) CLASSIC STRONG MOTION FILTER

DIG1, CH 2, (C2) THRESHOLD TRIGGER

DIG1, CH 3, (C3) CHANNEL EXTRACTOR

DIG1, CH 3, (C3) CLASSIC STRONG MOTION FILTER

DIG1, CH 3, (C3) THRESHOLD TRIGGER

DIG1, CH 4, (C4) CHANNEL EXTRACTOR

DIG1, CH 4, (C4) CLASSIC STRONG MOTION FILTER

DIG1, CH 4, (C4) THRESHOLD TRIGGER

VOTER

RECORDER

ALTUS EVT FORMAT DATA ARCHIVER

COMMAND CONSOLE

ROCK GPIO

STATUS SERVER

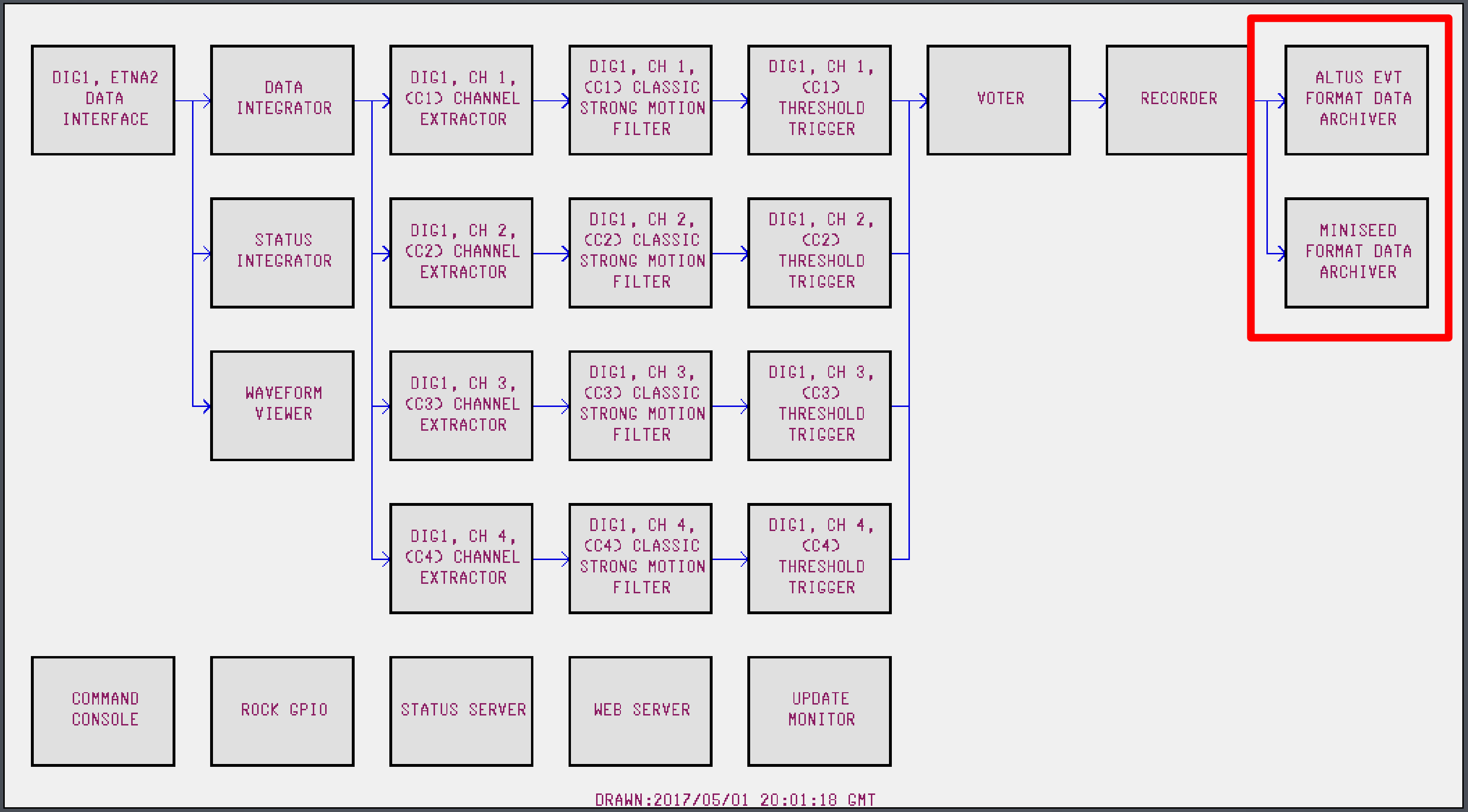
WEB SERVER

UPDATE MONITOR

DRAWN:2017/05/01 18:47:02 GMT

Examples$_7$

Recording in 2 Formats


DIG1, ETNA2 DATA INTERFACE
DATA INTEGRATOR
DIG1, CH 1, (C1) CHANNEL EXTRACTOR
DIG1, CH 1, (C1) CLASSIC STRONG MOTION FILTER
DIG1, CH 1, (C1) THRESHOLD TRIGGER
VOTER
RECORDER
ALTUS EVT FORMAT DATA ARCHIVER
STATUS INTEGRATOR
DIG1, CH 2, (C2) CHANNEL EXTRACTOR
DIG1, CH 2, (C2) CLASSIC STRONG MOTION FILTER
DIG1, CH 2, (C2) THRESHOLD TRIGGER
MINISEED FORMAT DATA ARCHIVER
WAVEFORM VIEWER
DIG1, CH 3, (C3) CHANNEL EXTRACTOR
DIG1, CH 3, (C3) CLASSIC STRONG MOTION FILTER
DIG1, CH 3, (C3) THRESHOLD TRIGGER
DIG1, CH 4, (C4) CHANNEL EXTRACTOR
DIG1, CH 4, (C4) CLASSIC STRONG MOTION FILTER
DIG1, CH 4, (C4) THRESHOLD TRIGGER
COMMAND CONSOLE
ROCK GPIO
STATUS SERVER
WEB SERVER
UPDATE MONITOR

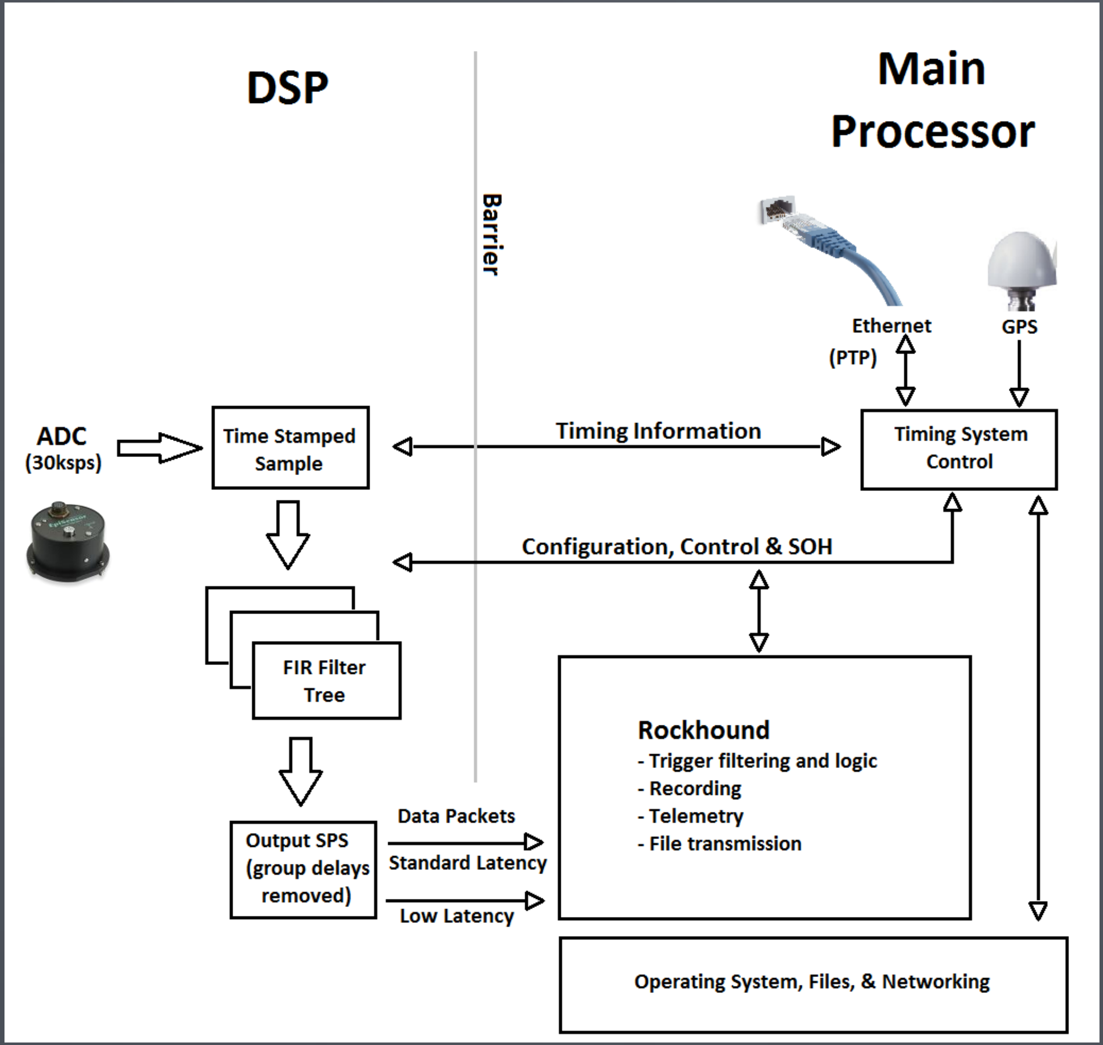
DRAWN:2017/05/01 20:01:18 GMT

Typical Rock Digitizer Data Flow$_1$

Who Does What?

- Time-critical processing is done in the DSP
  - Time-stamping of data at the input sample rate
  - FIR filtering to output sample rates
  - Initial data packaging
- Higher Level Features in Main processor
  - Triggering and Trigger Filtering
  - Output File Formatting
  - Telemetry
  - Configuration
  - User Interfaces
  - Peripherals and media

Typical Rock Digitizer Data Flow$_2$



Rockhound ORB

Rockhound’s ORB server can be RAM based or disk based, depending on requirements

It can serve out standard (1s) or low latency (0.1s) data as well as messages and status

The data can be accessed via orb2orb

Our implementation was written this way to be efficient in the digitizer environment on multiple platforms and processors

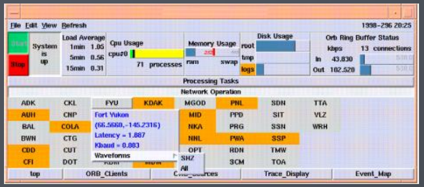
Rockhound ORB Status

Rockhound’s ORB Server produces pf/st packets.

These packets can be configured to include “pretty much” anything that Rockhound knows about.

This is done using a pfst.cfg file that can be found in the SMARTSDist folder.

Data available from Rockhound is basically everything reported with the “rtparams” command on the Rockhound Console (port 9900).



1998-296 20:25

System is up

Load Average 1min 1.05 5min 0.56 15min 0.31

Cpu Usage cpu#0 71 processes

Orb Ring Buffer Status kbps 13 connections In 43.830 Out 102.520

Fort Yukon
(66.5660,-145.2316)
Latency = 1.887
Kbaud = 0.883

pf/st Status Packet$_1$

```
#=======================
# Status conversion file
#=======================
#
# Used to convert system status information into data elements useable by
# Antelope.
#
# Each line contains one variable as follows:
#
# pfstvarname rhvarname flags
#
# Where:
# - pfstvarname = Variable name inserted into the pf/st packet
# - rhvarname = Variable name found in Rockhound runtime parameters
# - Flags:
#     *Scale = Multiplier scale factor
#     Fx = Formatting of digits (number of places)
#     >x = Only display if greater than x
#     >85:t >95:l = Conversion of values to strings (if greater than x)
#     Drhvar = Delta (difference) vs another Rockhound parameter
```

pf/st Status Packet$_2$

```
#    P = This is a persistent parameter
#    Nx = Display as 'x' if null
#    H = Hardcoded parameter
aa        dig1.LocalGPSAntCurrent      *0.001          F3
cld       dig1.LocalTcxoDrift          *0.001
clq       dig1.LocalClockQuality       >0:?            >85:t          >95:l
clt       dig1.LocalGPSLockChanged     *0.001          F3
da        dig1.LocalRockCurrent        *0.001          F3
dg        DataIntegrator.NMissingGroups
dh        dig1.LocalHumidity
dlt       dig1.LocalTimeSinceDataArrival    *0.001     F3
dt        dig1.LocalTemperature
dv        dig1.LocalRockDCVolts
elev      dig1.LocalGPSAltitude             *0.001     F3
esn       dig1.LocalESerialNumber
gp1       gp1                          H
gp24      gp24                         H
lat       dig1.LocalGPSLatitude
lcq       dig1.LocalClockQuality
lon       dig1.LocalGPSLongitude
```

pf/st Status Packet$_3$

m0	dig1.bd0.LocalMassPos1	*0.001	F3	
m1	dig1.bd0.LocalMassPos2	*0.001	F3	
m2	dig1.bd0.LocalMassPos3	*0.001	F3	
nc	dig1.NPhysicalChannels			
nrb	NStartups	P		
nr24	nr24	H		
rtm	rtm	H	*0.001	F3
sn	dig1.LocalSerialNumber			
trb	SMARTS.SystemStartTime	*0.001		
vco	dig1.LocalTcxoDAC			

Runtime Parameters (port 9900)

```
> rtparams
AltusEVTStorage=/data/events/
AltusEvtArchiver.Type=AltusEvtArchiver
CommandConsole.Type=CommandConsole
…
dig1.SensorRange=2g
dig1.ch1.Altitude=0
dig1.ch1.Azimuth=0
dig1.ch1.Damping=0.7
dig1.ch1.DetriggerVotes=1
dig1.ch1.EpiCalCoil=0.0599
dig1.ch1.EpiGain=2
dig1.ch1.EpiRange=2
dig1.ch1.FullScale=2.5
dig1.ch1.FullScaleADCCounts=8388608
dig1.ch1.Gain=1
dig1.ch1.Id=C1
dig1.ch1.NaturalFreq=204.0
```