Antelope
developments for 5.5 release

March, 2015
Antelope User Group Meeting
OGS, Udine, Italy

BRTT
BOULDER REAL TIME TECHNOLOGIES, INC.

BOULDER
121·RUH
NISSAN

- Licensing
- Major new development
- Outsourcing

Licensing – Current State

- Explain Antelope IP and node licensing
  - IP licenses enabled by particular WAN IP addresses
  - Node licenses enabled by particular hardware
- IP-based licensing appears to be stable and secure, although still presents problems in hypersecure IT environments
- Node licensing is simple and works well for Apple systems
- Node licensing is problematic for Linux systems
- Node licensing is currently unavailable for virtual hosts

Licensing – Looking Forward

- IP licensing will remain the same
- Apple node licensing will remain the same
- Linux node licensing for CentOS 7+ will require a new licensing daemon that will replace `hald` used in CentOS 6.x
- The new licensing daemon could come from a commercial source, such as Safenet's HASP USB dongle
- The new licensing daemon could be written by BRTT
- Almost certainly the new Linux licensing daemon will require root permission for installation
- The USB dongle option could possibly enable node licensing for virtual hosts

Licensing – Contributed/GPL

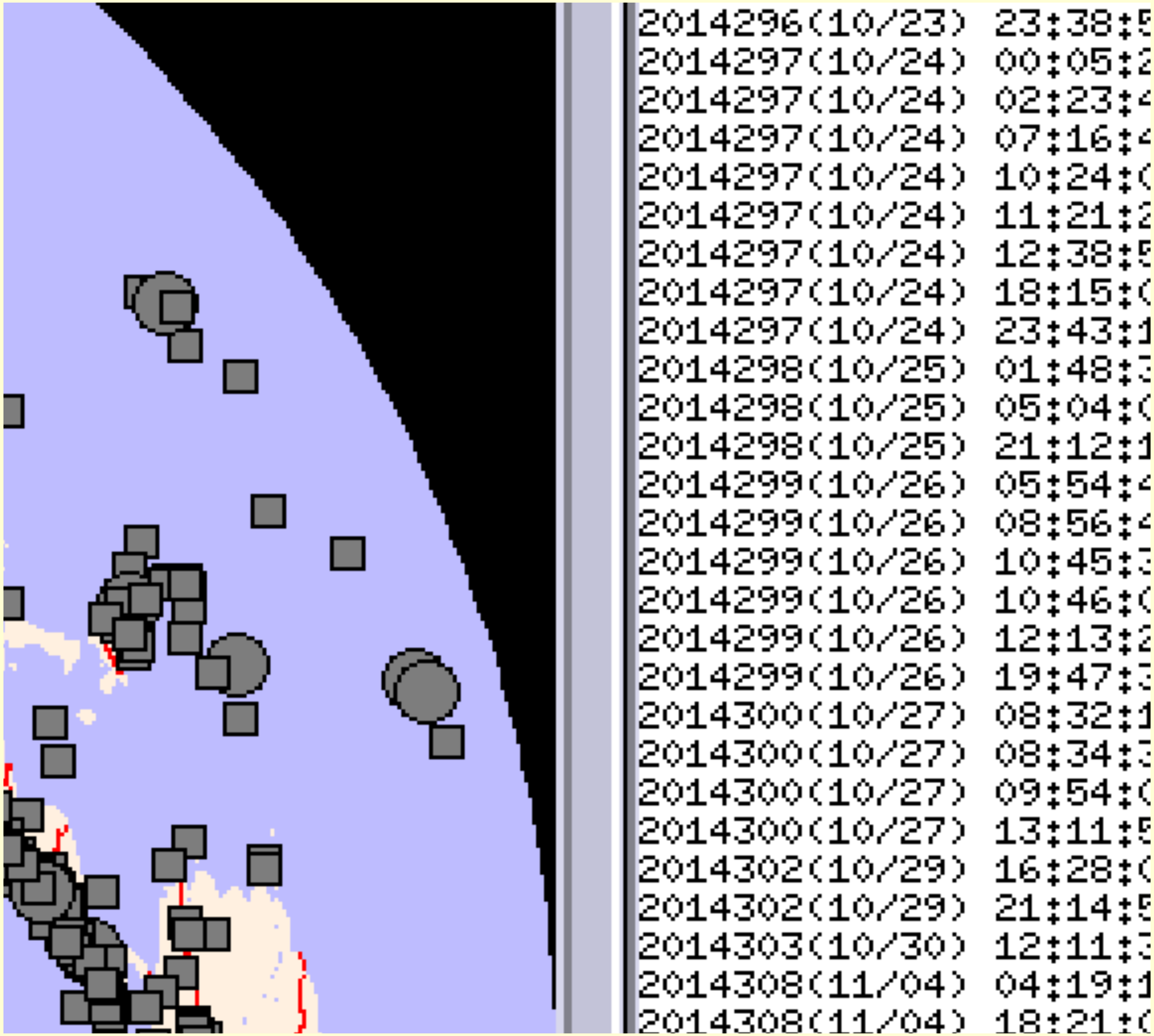
- Explain evolution of "open source" licensing
- BRTT core software along with anything BRTT releases cannot make use of GPL licensed external software (such as ObsPy)
- BRTT will follow a strict licensing declaration protocol in 5.5
- Some or all of contrib software will probably be put into a separate distribution ISO that will come from the AUG web site or something in the cloud
- Note that companies like Mathworks have this same problem

How do we prioritize major new developments?

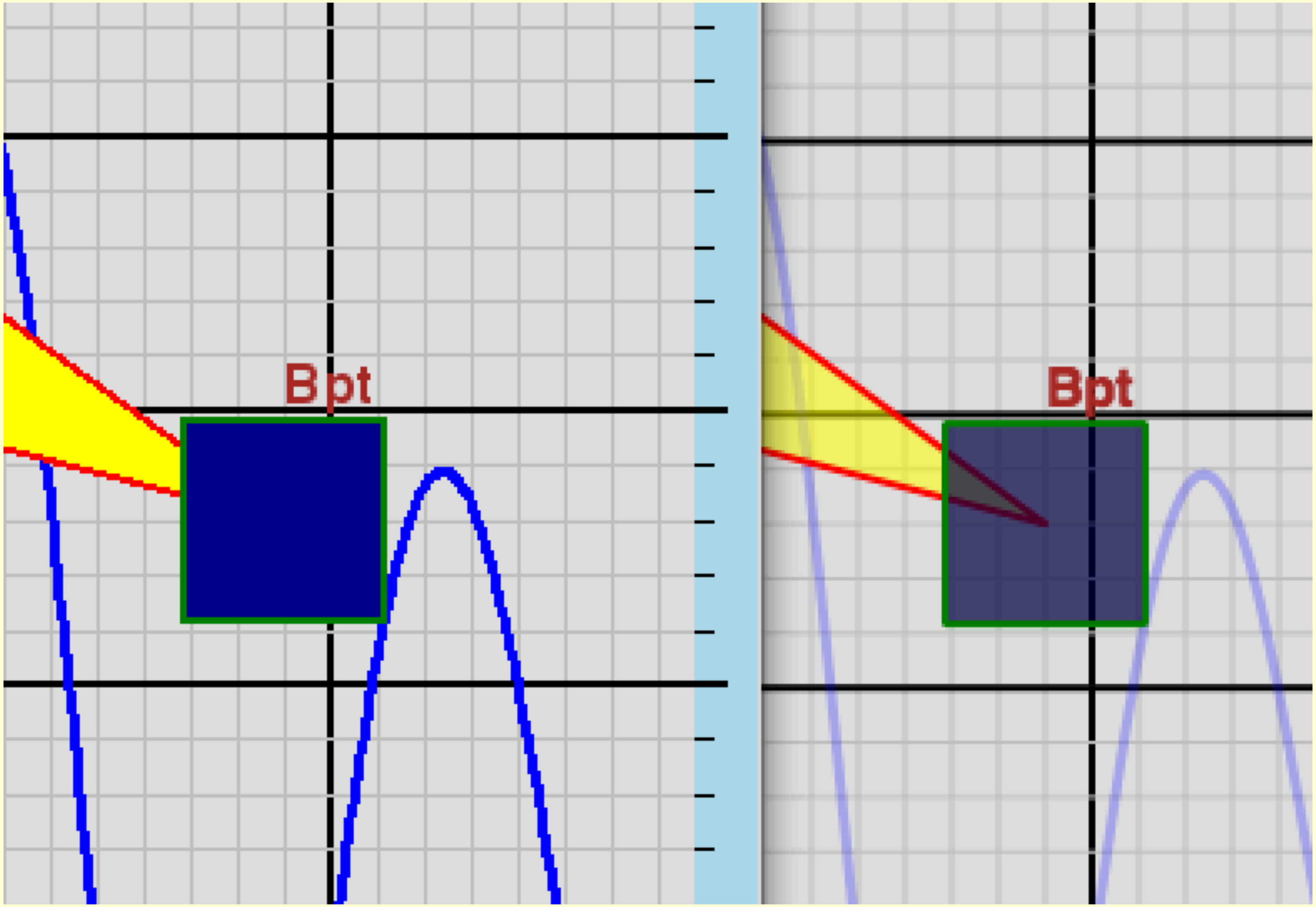
- Complex balancing of desires and capabilities
- Desires come from us at BRTT and our user community (which is why these meetings are so important to us)
- Prioritization factors:
  - How many users want something? Is it important for internal functionality?
  - Does a new development make sense now? (opportunity, availability, fits within Antelope software model, etc.)
  - Does a new development fit within BRTT's overall development constraints?
  - Is the new development properly specified?
  - Is it something BRTT is competent to do or can be outsourced?
  - How hard is it to do?
  - Suggest making a spreadsheet during this meeting – we can evolve this and could be an important decision making tool in the future

Major New Development – Modern Display Graphics and GUIs

- Display and Interactive/what we have now
  - Fundamentally X11/TK based
    - 30+ year old technology
    - Largely unsupported
    - Not likely to evolve
    - Lowest common denominator (low level APIs)
    - Does not fully support modern graphics capabilities
    - Not portable to mobile devices
    - Provides UNIX cross platform compatibility
    - Provides remote graphics/interaction forwarding
    - Used X11/Tk as perl and python extensions
    - Developed BRTT custom extensions on X11/Tk



```
2014296(10/23)  23:38:5
2014297(10/24)  00:05:2
2014297(10/24)  02:23:4
2014297(10/24)  07:16:4
2014297(10/24)  10:24:0
2014297(10/24)  11:21:2
2014297(10/24)  12:38:5
2014297(10/24)  18:15:0
2014297(10/24)  23:43:1
2014298(10/25)  01:48:3
2014298(10/25)  05:04:0
2014298(10/25)  21:12:1
2014299(10/26)  05:54:4
2014299(10/26)  08:56:4
2014299(10/26)  10:45:3
2014299(10/26)  10:46:0
2014299(10/26)  12:13:2
2014299(10/26)  19:47:3
2014300(10/27)  08:32:1
2014300(10/27)  08:34:3
2014300(10/27)  09:54:0
2014300(10/27)  13:11:5
2014302(10/29)  16:28:0
2014302(10/29)  21:14:5
2014303(10/30)  12:11:3
2014308(11/04)  04:19:1
2014308(11/04)  18:21:0
```


Bpt
Bpt

What are the differences?

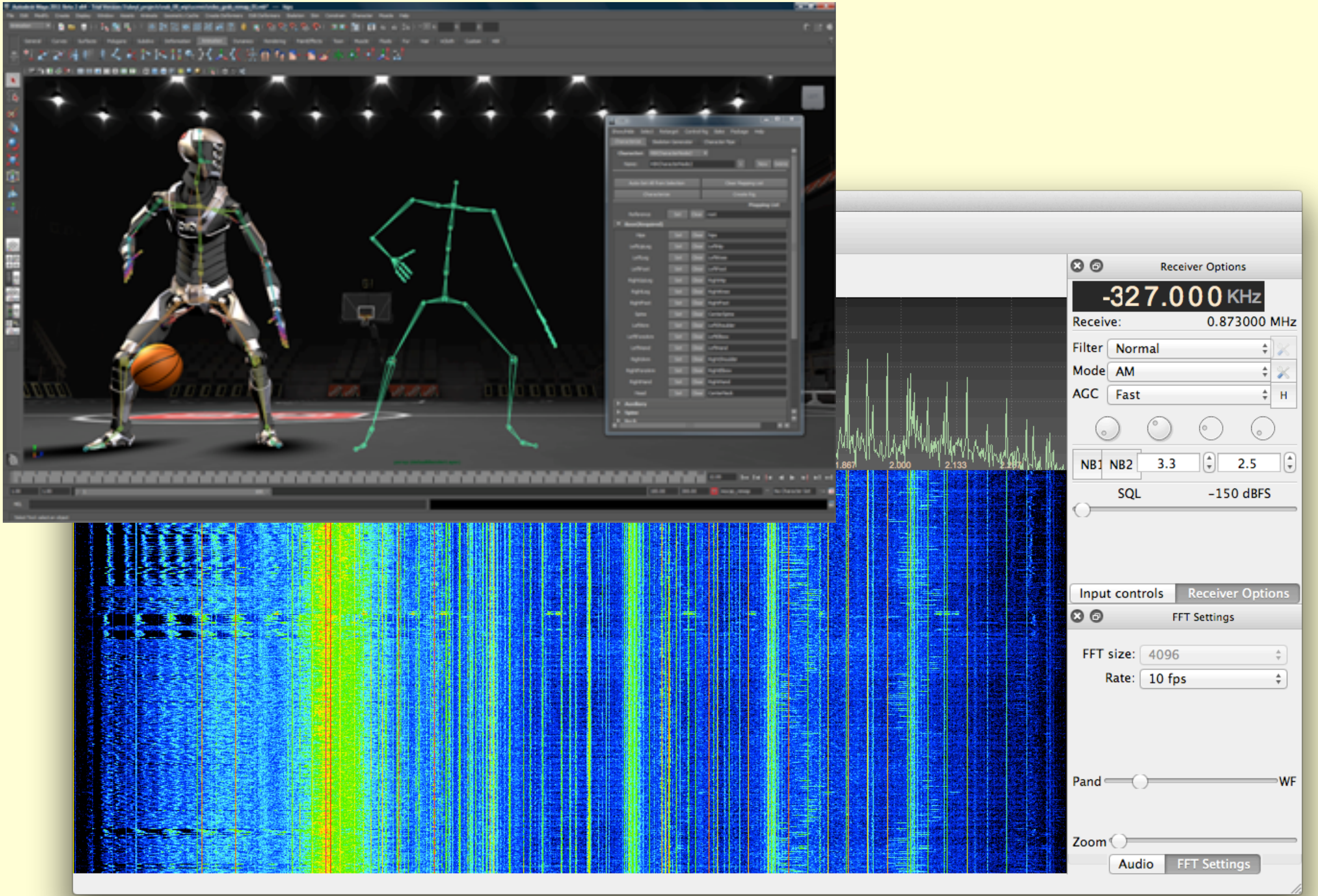
- Spatial antialiasing for fonts and graphics
- Ability to efficiently blend pixels

Modern Display Graphics and GUIs

- What do we want?
  - Modern cross platform graphics/display middleware
    - Fully supported (not by BRTT)
    - Fully supports modern graphics hardware capabilities
    - Commercially licensed
    - Can be built on UNIX and mobile architectures
    - Has large use base
    - Supports web-base applications
    - Provides high level APIs

Modern Display Graphics and GUIs

- What are we going?
  - Qt
    - Cross platform API (MacOSX, LINUX, Windows, iOS, Android, Windows Mobile)
    - Commercially supported and licensed (Digia)
    - High level support for modern graphics hardware (fonts, spatial antialiasing, alpha blending, 3D rendering, etc.)
    - Very large user base (Nokia, KDE, Android apps, embedded devices) plus sophisticated extensions such as Marble
    - QTWebkit and QTWebsockets plus XML interpreter
    - Up to OpenGL API levels



Receiver Options

-327.000 KHz

Receive: 0.873000 MHz

Filter: Normal

Mode: AM

AGC: Fast

NB1 NB2 3.3 2.5

SQL -150 dBFS

FFT Settings

FFT size: 4096

Rate: 10 fps

BRTT

March 2015

What is Qt?

- Graphics/Interaction middleware
- C++ API with ~500 classes
- High performance at various levels
- High functionality at various levels
- Cross platform API with common application code base for MacOSX/Cocoa, MaxOSX/Xquartz/X11, Linux/X11, iOS, Android, Windows
- Both GPL and commercially licensed through Qt Company

How does Qt impact our users?

- Not at all, for our users who do not plan on doing their own development of graphics/interaction software
- BRTT's Qt commercial license allows us to develop software using commercial Qt and to distribute these developments free of license fees to our users
- BRTT's Qt commercial license allowing software development does not extend to our users
- There is a parallel freely available GPU licensed Qt middleware that our users could use for their developments as long as our users can abide with the GPU license terms

How will BRTT make use of Qt?

- BRTT will stop all graphics/GUI development that uses X11/Tk. This includes the TCL, perl and python extensions we have used and developed in the past.
- Starting with 5.5, new graphics/GUI software will be developed only using Qt
- Although there is a dual GPU/commercial PyQt python extension library for Qt, BRTT will not use PyQt for the 5.5 release (we have experimented with making our own version of PyQt)
- New BRTT developed graphics/GUI software will be written in c++

Qt developments for 5.5

- New `buplotqt` library that introduces buplot extensions into Qt
- New `dbe`-like prototype
- New `dbevent`-like prototype
- Live demo

Qt developments for 5.5

- Complete rewrite of BRTT map display software
- Support for continuously scalable display transformations of image data such as NASA’s Bluemarble earth image data
- High performance map projection transformations through threading
- New BRTT Map Data (bmd) format that supports multiple resolution and tiled image and vector data in both native compressed and uncompressed formats
- Large map files, such as Bluemarble bmd file, will be separately downloaded

Qt development path

- Next major task is implementing new trace plotting extensions used in `orbrtd` and for future version of `dbpick`
- We have our own prototype extensions for python and may develop that further depending on licensing issues (alternative is use of PyQt)
- Need to start backfilling old GUI and display codes, such as `dbe`
- Extend to web client applications

Outsourcing

- BRTT staff will be focusing on infrastructure development
- BRTT will be outsourcing “app” software development, such as moment tensor, focal mechanism, Brune spectra, various format conversions, etc.
- We welcome any suggestions about our outsourced development including what should be developed and recommendations for who we should engage to do the development