sunpy

A Community Python Library for Solar Physics

Andrew. R. Inglis[1]

on behalf of **The SunPy Community**

1. NASA Goddard Space Flight Center / Catholic University of America

Solar Orbiter Data Analysis Working Group

10th September 2014

Why Python?

- Python is very versatile with a large amount of functionality. Also widely not only in **science**, but in many other fields, such as **IT security, video game design, web development** (e.g. Google, YouTube).

- **Easy to interact with external software**, such as SQL databases, C libraries, websites, bash/csh shell. Standard tools for HTML/XML parsing.

- **Modular design** – easy to install only what you need. Dependencies are easily managed.

Why Python?

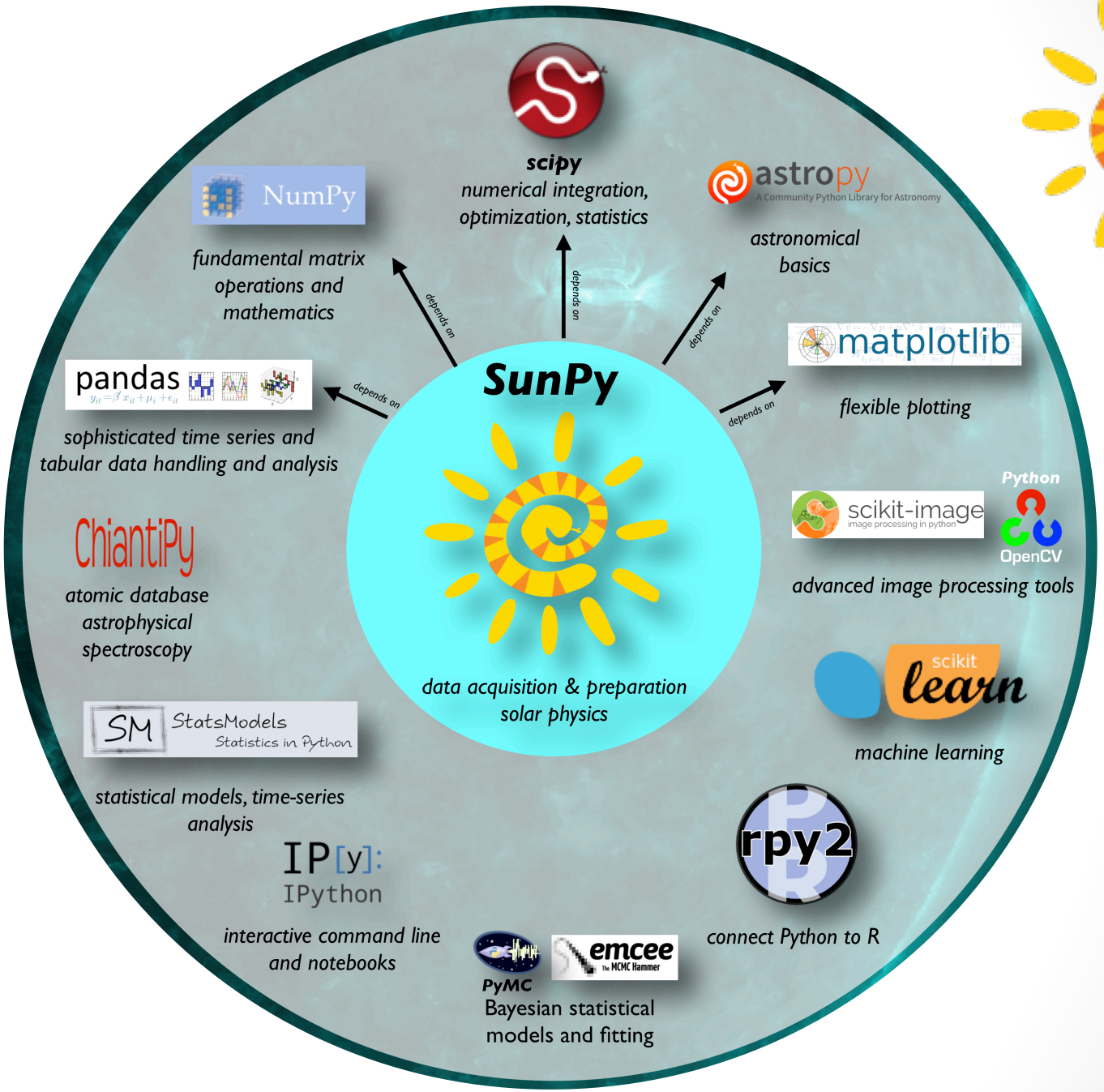
- Python is free and a widely used language in many science-related fields, including 'big data', finance, computing, as well as physics.

- Numerous scientific packages already exist that can be leveraged:
  - NumPy - *basic scientific programming functionality*
  - SciPy - *scientific tools, fitting, statistics, errors*
  - AstroPy - *astrophysical tools, coordinates, units*
  - SpacePy – *space-based tool collection*
  - Pandas - *time series*
  - Matplotlib - *plotting*
  - PyMC - *Monte-carlo procedures, Bayesian analysis*

- Python has an ***active community***

What is SunPy?

- ***Free, open source*** package for solar physics written in Python.
- Provides high level functionality for accessing solar physics data and performing analysis of that data.
- Community led – SunPy development is led ***by solar physicists,*** ensuring the needs of the solar community can be met.
- Fully version controlled using ***Git*** and ***GitHub*** – all software is downloadable online, every change is documented, anyone can contribute.


scipy
numerical integration, optimization, statistics
NumPy
fundamental matrix operations and mathematics
astropy
A Community Python Library for Astronomy
astronomical basics
depends on
depends on
depends on
depends on
depends on
SunPy
data acquisition & preparation solar physics
pandas
sophisticated time series and tabular data handling and analysis
matplotlib
flexible plotting
ChiantiPy
atomic database astrophysical spectroscopy
scikit-image
image processing in python
Python
OpenCV
advanced image processing tools
scikit learn
machine learning
SM
StatsModels
Statistics in Python
statistical models, time-series analysis
IP[y]:
IPython
interactive command line and notebooks
rpy2
connect Python to R
PyMC
emcee
The MCMC Hammer
Bayesian statistical models and fitting

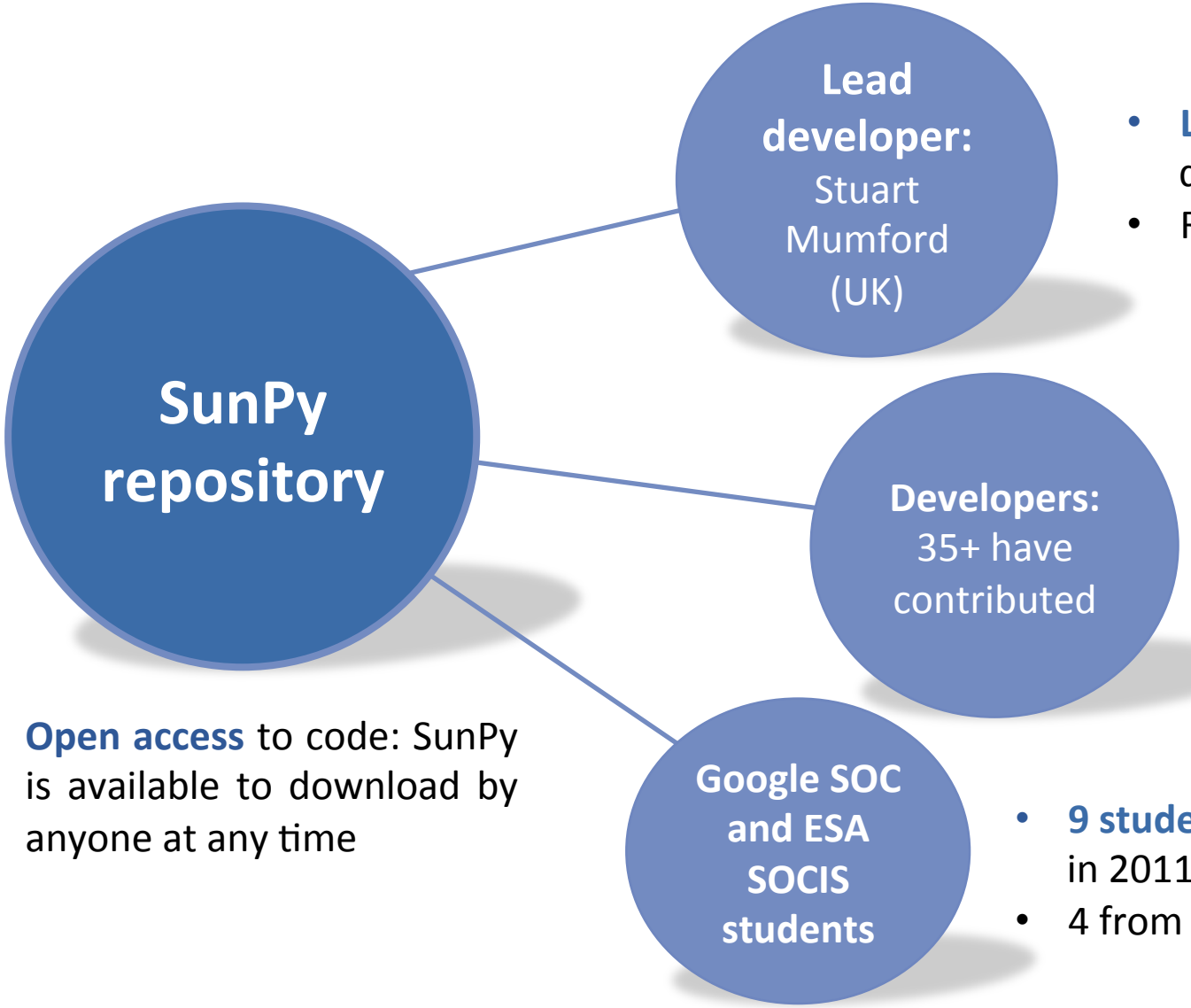
SunPy organisation

The SunPy Board		
Steven Christe	Chair	USA
David Perez-Suarez	Vice-Chair	SA
Andrew Inglis	Secretary	USA
Stuart Mumford	Member	UK
Albert Shih	Member	USA
Juan Carlos Martinez	Member	USA
Tom Robitaille	Member	GER
Jack Ireland	Member	USA
Russel Hewett	Member	USA

Makes decisions regarding the **overall direction** of the SunPy project, as well as **maintaining SunPy resources**, such as websites, repositories.

Interacts with external organisations, such as Google Summer of Code (GSOC), ESA Summer of Code in Space etc.

SunPy organisation



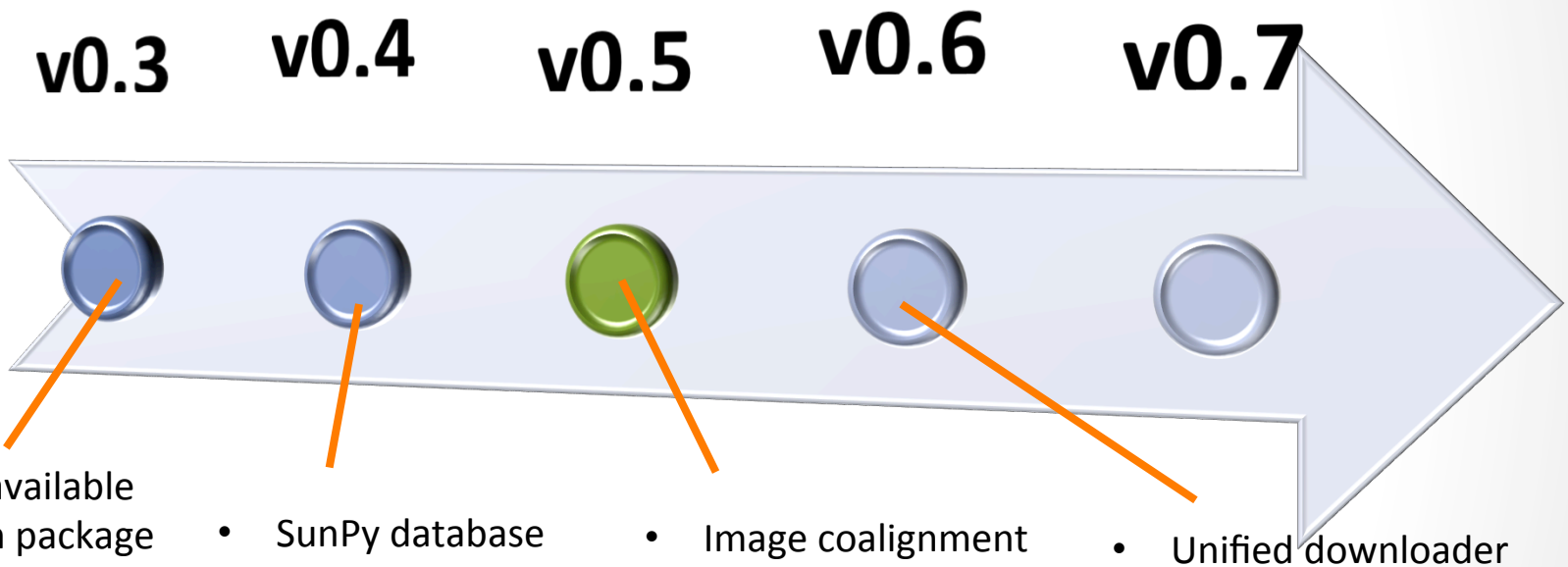
- **Leads** day-to-day code design and implementation
- Reports to board

- **9 new contributors** to SunPy 0.5 release (June 2014)
- International developer community
- **Anyone** can be a developer!

Open access to code: SunPy is available to download by anyone at any time

- **9 students** since SunPy founded in 2011
- 4 from GSOC in 2014!

SunPy Roadmap



- SunPy available through package installers
- Map refactored
- IO improvements – FITS reading
- Spectrum made consistent with Map, Lightcurve.

- SunPy database
- Major docs upgrade
- Introduced Astropy constants
- Some IRIS support
- MapCubeAnimator class introduced

- Image coalignment
- Map rotate improvements
- GOES temp and emission measure
- Added NoRH and NOAA summary data to lightcurve.

- Unified downloader
- Lightcurve refactor
- Astropy Quantities in functions
- Fermi/GBM lightcurve support
- WCS coordinate integration with Map

SunPy version control

sunpy / sunpy

Unwatch 22 | Unstar 148 | Fork 95

August 10 2014 - September 10 2014

Period: **1 month**

Overview

25 Active Pull Requests	**20** Active Issues

14 Merged Pull Requests	**11** Proposed Pull Requests	**4** Closed Issues	**16** New Issues

Excluding merges, **16 authors** have pushed **19 commits** to master and **222 commits** to all branches. On master, **19 files** have changed and there have been **246 additions** and **39 deletions**.

14 Pull requests merged by **6** people

Merged #1174 **Add changelog entry for STEREO** 2 days ago

Merged #1170 **[Review]: Add STEREO HI colour maps and a HIMap class.** 2 days ago

Merged #1167 **Sphinx 1.2.2 in Travis** 5 days ago

Contributing to SunPy

Anyone can contribute to SunPy, but **guidelines are in place** to ensure that new code meets quality standards.

Unit test coverage

- Tests ensure code works as intended
- Tests insure against future code changes

Pep8 compliance

- Good coding practice. Good code readability
- Well commented, documented code

Peer reviewed

- New code is peer-reviewed by other developers
- Code only accepted after outstanding issues are addressed

All new code is thoroughly reviewed by the **developer community** and the **Lead Developer**. Automated services (Travis, Coveralls) assist with the testing of new potential contributions.

SunPy basics

There are three main types of data in solar and astrophysics. These are images, time-series, and spectra. Accordingly, SunPy contains three object classes designed to manipulate these data types:

Map – for individual images and sequences of images.

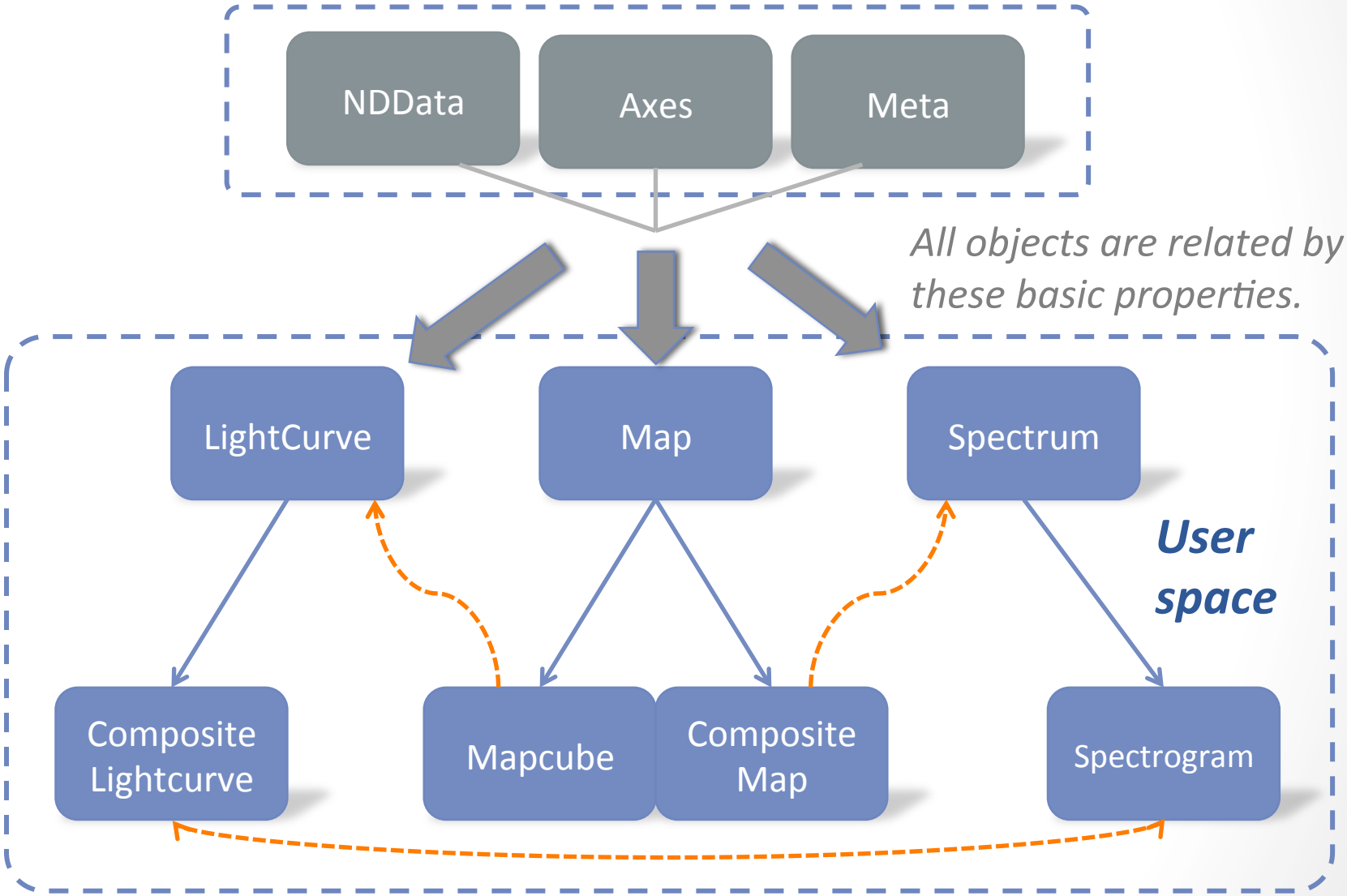
Lightcurve – for 1D time series data.

Spectrum – for 1D and 2D data in frequency and time.

These object provide a **uniform, consistent** way to manipulate the observational data, regardless of the source.

Additionally, SunPy provides modules for accessing data from the ***VSO***, ***HEK***, ***HELIO*** and ***Helioviewer*** systems, as well as ***world coordinate system (WCS)*** transformations, ***units***, ***solar ephemeris***, ***solar constants*** and much more.

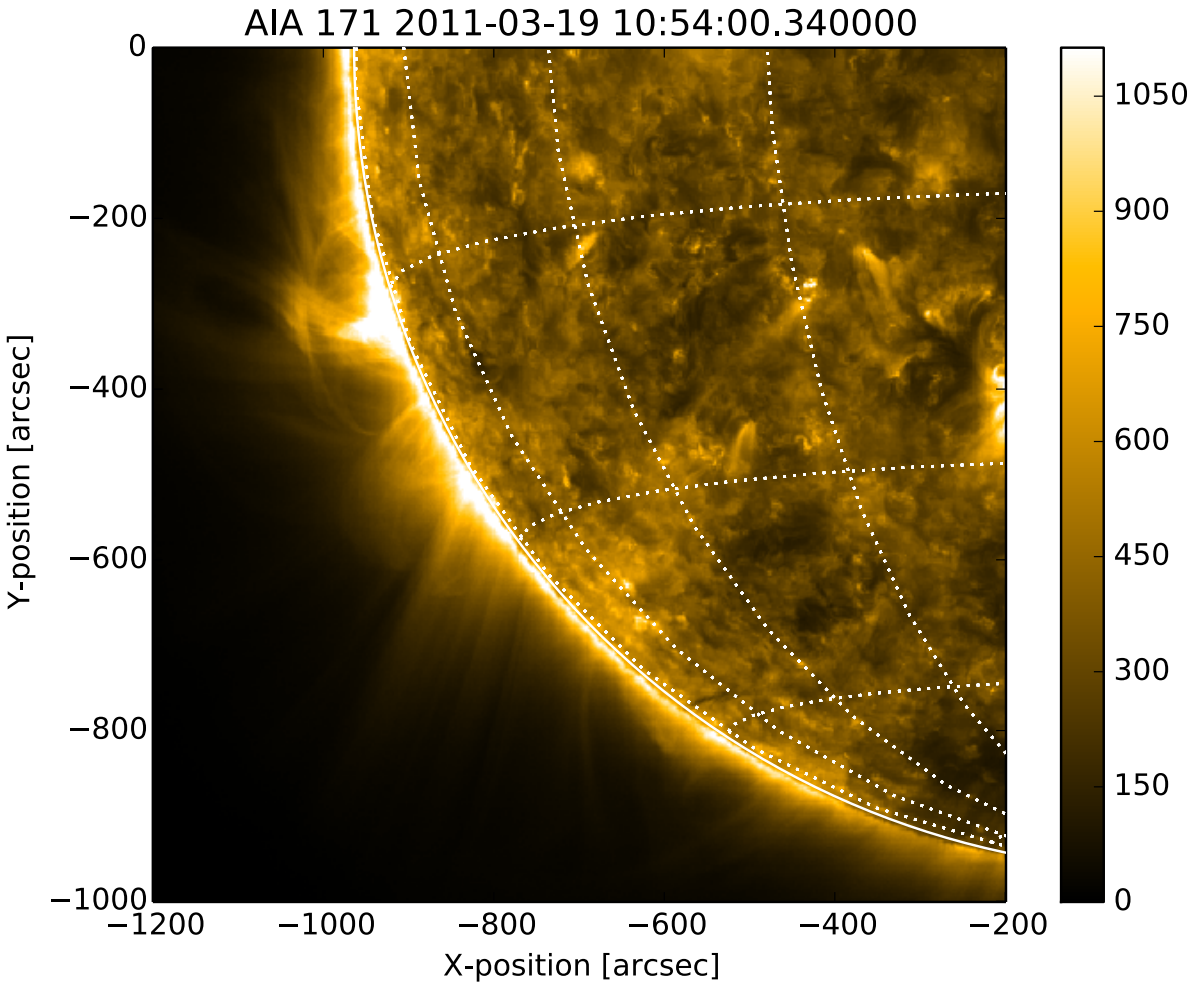
SunPy basics: architecture



Map

The map class is designed for holding 2D image data, such as solar images.

A list of maps can also be placed into a **Mapcube object**. Operations (e.g. coordinate transformation) can then be performed on the entire object.

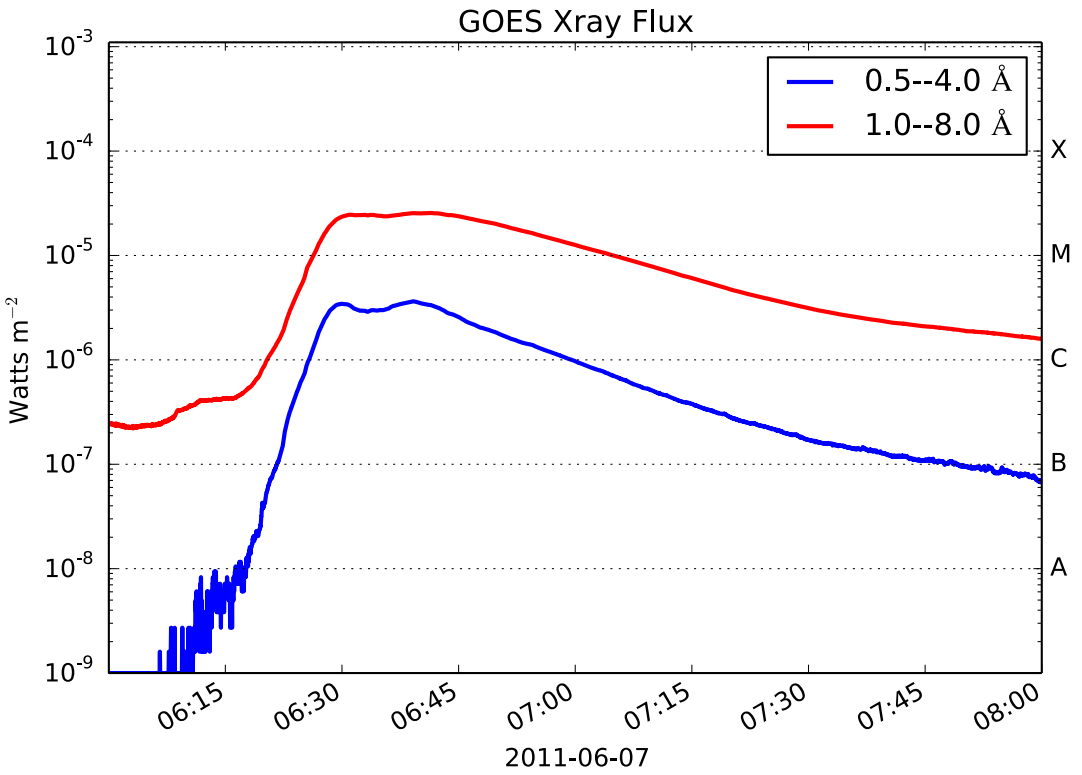



```
>>> import sunpy.map
>>> aiamap = sunpy.map.Map("AIA20110319_105400_0171.fits")
>>> smap = aiamap.submap([-1200, -200], [-1000, -0])
>>> smap.peek(draw_grid=True)
```

LightCurve

The LightCurve class is used for 1D time series data. Data columns are mapped to a global index column.

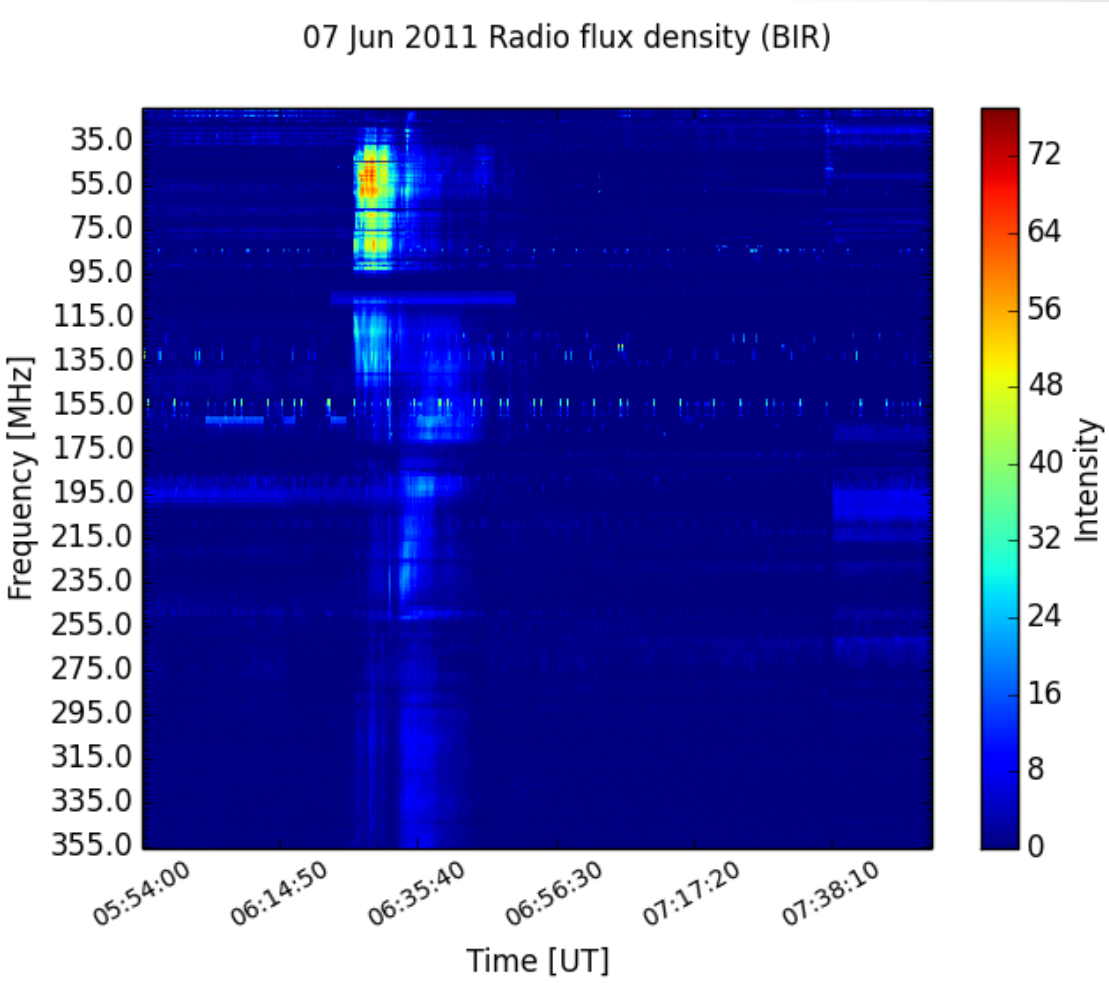
Lightcurves can easily be concatenated and truncated.




```
>>> from sunpy import lightcurve
>>> from sunpy.time import TimeRange
>>> tr = TimeRange("2011-06-07 06:00", "2011-06-07 08:00")
>>> goes = lightcurve.GOESLightCurve.create(tr)
>>> goes.peek()
>>> print('The max flux is ' + str(goes.data['xrsb'].max()) +
...       ' at ' + str(goes.data['xrsb'].idxmax()))
The max flux is 2.5554e-05 at 2011-06-07 06:41:24.118999
```

Spectrum

The Spectrum data type is used for both spectrographic 2D data (time and frequency) or 1D spectral data (e.g. counts vs energy.)




```
>>> from sunpy.spectra.sources.callisto import CallistoSpectrogram
>>> tstart, tend = "2011-06-07T06:00:00", "2011-06-07T07:45:00"
>>> callisto = CallistoSpectrogram.from_range("BIR", tstart, tend)
>>> callisto_nobg = callisto.subtract_bg()
>>> callisto_nobg.peek(vmin=0)
```

SunPy and Solar Orbiter

	Instrument	Data type	File format
In situ	Energetic Particle Detector (EPD)	***LightCurve***	***Ascii***
	Magnetometer (MAG)	***LightCurve***	***Ascii, CDF***
	Radio and Plasma Waves (RPW)	***LightCurve***	***Ascii, CDF***
	Solar Wind Plasma Analyser (SWA)	***LightCurve***	***Ascii, CDF***
Remote sensing	Extreme Ultraviolet Imager (EUI)	***Map***	***FITS***
	Coronograph (METIS)	***Map***	***FITS***
	Polarimetric and Helioseismic Imager (PHI)	***Map***	***FITS***
	Heliospheric Imager (SoloHI)	***Map***	***FITS***
	SPICE	***Spectrum***	***FITS***
	X-ray imaging spectrometer (STIX)	***Map, Spectrum***	***FITS***

SunPy and Solar Orbiter

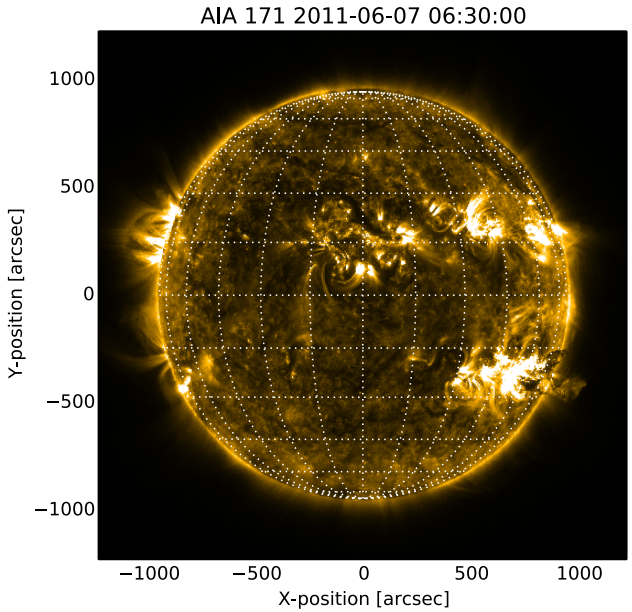
- **Ascii and CDF file support is available**. CDF readers are available in SciPy.
- SunPy integration with CDAWeb is planned for **easy data access** to heliospheric data**.**
- **SunPy Lightcurve object** provides time series analysis tools for in-situ data. **Map object** provides convenient framework for remote sensing images.
- Actively working on integrating **coordinate awareness and transform capability** into SunPy objects. This enables easier analysis of joint data sets.

World Coordinate System (WCS)

Standardized coordinate systems for solar data (Thompson 2006). Full coordinate information in FITS headers. WCS transformations are available in SunPy:

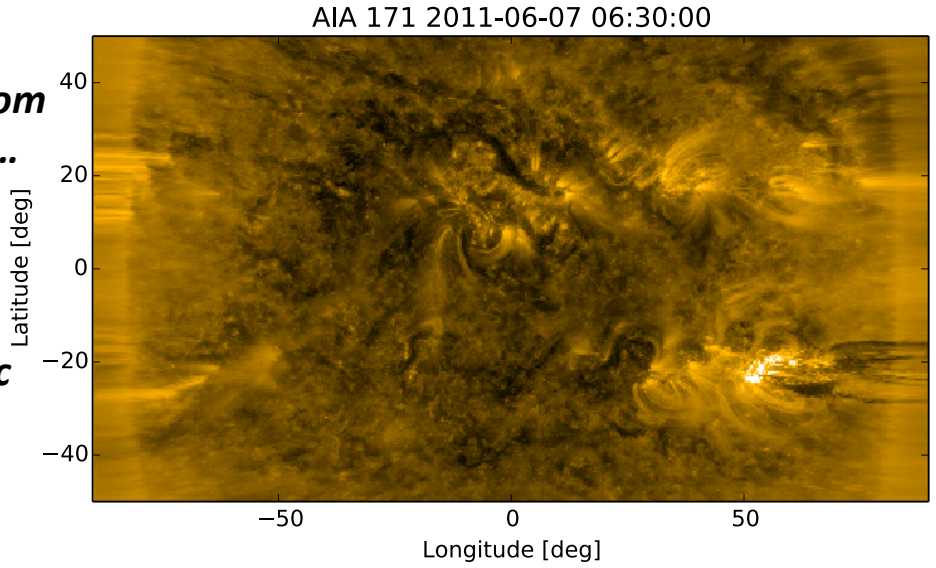
```
>>> from sunpy import wcs
>>> wcs.convert_hg_hpc(10, 53)
(100.49244115330731, 767.97438321917502)
```

We can use this to transform Maps:



For example, from Helioprojective…

to Heliographic



Quantities and Units

SunPy is moving to using **AstroPy Quantities**. These attach the dimensions of the unit to the value itself.

```
>>> import astropy.units as u
>>> angle_deg = 60 * u.deg
>>> print angle_deg
60.0 deg
>>> print 'The value is: ' + str(angle_deg.value)
The value is: 60.0
>>> print 'The units are: ' + str(angle_deg.unit)
The units are:deg
```

Convenient to change units.

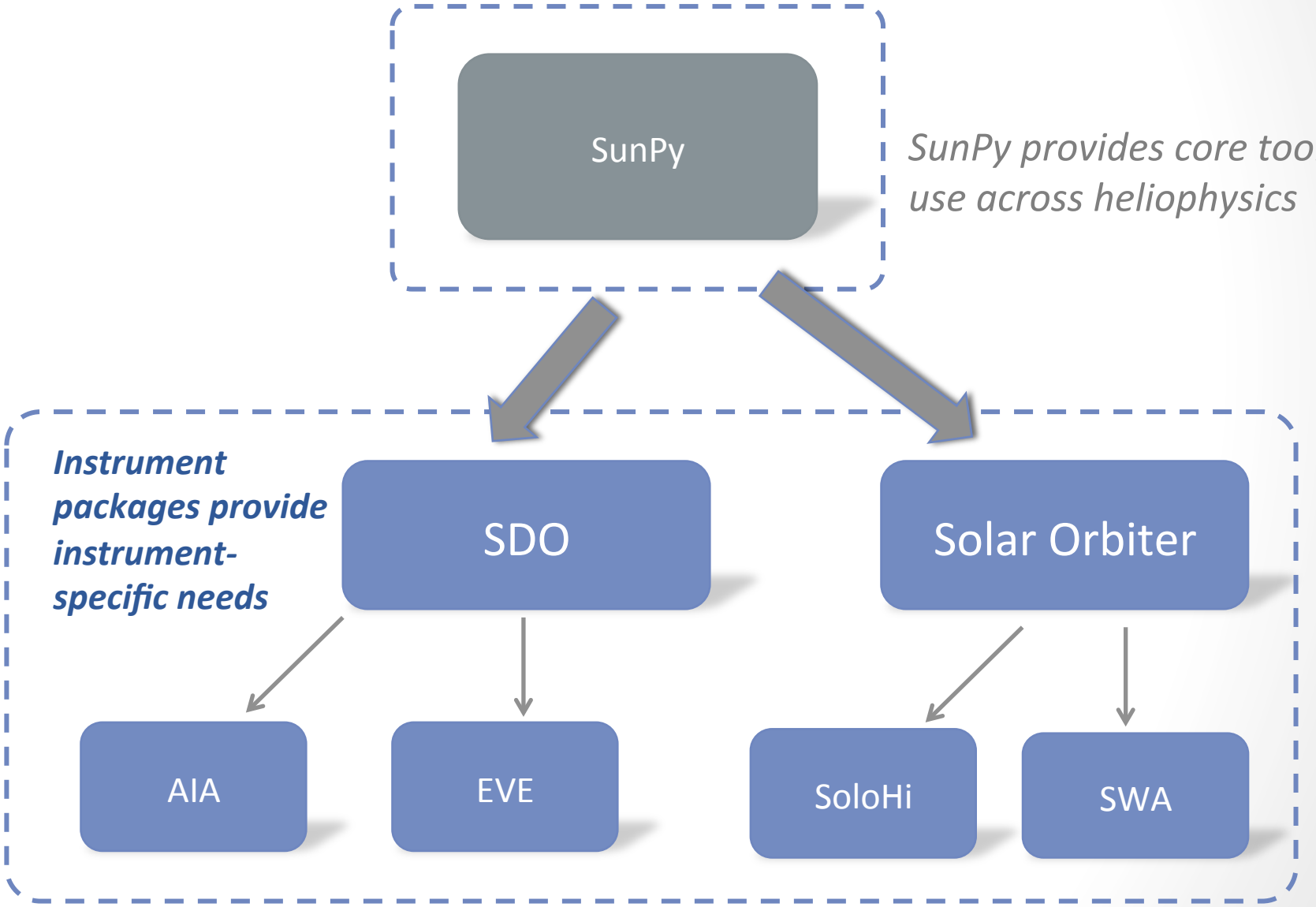
Unit changes are automatically handled during mathematical operations (multiplication, division).

```
#now convert to radians

>>> angle_rad = angle_deg.to('rad')
>>> print angle_rad
1.0471975512 rad
```

```
#multiply with some other units
>>> d = 10 * u.meter
>>> combo = angle_rad * d
>>> print combo
10.471975512 m rad
```

SunPy future vision



In Summary…

- SunPy is steadily developing, with input from many colleagues internationally. Development is being substantially aided by success in Google Summer of Code and ESA Summer of Code in Space programs.

- SunPy is available now - we are actively seeking **community feedback**!

- Ultimately, we want to work together with future heliophysics missions. One future scenario is the **development of instrument-specific software packages that are affiliated with SunPy**.