

# **Europython 2006**

**Monday 03 July 2006 - Wednesday 05 July 2006**

**CERN, Geneva**

## **Programme**

# Table of contents

Monday 03 July 2006	4
Registration	4
Agile Development	4
Python Language and Libraries	4
Refereed Papers	5
Web Frameworks	5
Python in Science	5
Agile Development	6
Python Language and Libraries	7
Python in Science	7
Web Frameworks	8
Agile Development	8
Python Language and Libraries	9
Python in Science	9
Web Frameworks	10
Games	10
Python Language and Libraries	11
Python in Science	12
Keynote	12
Tuesday 04 July 2006	13
Business and Applications	13
Python Language and Libraries	13
Refereed Papers	14
Web Frameworks	14
Business and Applications	15
Python Language and Libraries	15
Python in Science	16
Web Frameworks	17
Business and Applications	17
Python Language and Libraries	17
Web Frameworks	18
Lightning Talks	19
Keynote	19
Wednesday 05 July 2006	20
Business and Applications	20
Refereed Papers	20

Teaching	20
Web Frameworks	21
Agile Development	21
Misfits	22
Teaching	22
Web Frameworks	23
Agile Development	23
Misfits	24
Web Frameworks	25
Lightning Talks	25

# Monday 03 July 2006

## Registration: Building 33 (03 July 08:00-09:00)

## Agile Development - 40-SS-C01 (03 July 09:00-10:40)

time [id] title

09:00	<b>[27] Ajax, state of the art (00h30')</b> <i>Speaker: Mr. ZIADé, Tarek</i> Ajax has played an important role over the last months on improving the user experience over web interfaces. Besides all the buzz around Ajax and Web 2.0 related technologies, numerous Javascript libraries have started to consolidate some coding patterns and to provide usable features. This presentation is a state of the art of Ajax related technology, and presents what web developers can benefit from it, with considerations about accessibility (WAI) and ergonomics matters, but also with a focus on Javascript TDD (Test Driven Development) and continuous integration in a Python environment. Last but not least, the presentation will focus on Ajax uses cases for ECM (Entreprise Content Managment) and CMS (Content Managment System) systems, and implementations examples, taken from Nuxeo CPS. References: <a href="http://blogs.nuxeo.com/sections/blogs/tarek_ziade/archive_view?category=AJAX">http://blogs.nuxeo.com/sections/blogs/tarek_ziade/archive_view?category=AJAX</a> <a href="http://cps-project.org">http://cps-project.org</a>
09:35	<b>[110] Testing web applications with HyperTest (00h30')</b> <i>Speaker: Mr. WESDORP, Guido</i> As web applications grow increasingly complex, and the logic of those applications is located on the client more often, the need for good JavaScript testing tools is growing. There are some products available, but the quality is usually not satisfactory: most tools are inflexible, messy, hard to integrate, badly written or incomplete. HyperTest tries to solve those issues. HyperTest is a set of JavaScript libraries that allows testing all aspects of a web application: there's a unit test library to test JavaScript APIs, a functional testing library for user interfaces and a library for testing REST APIs by sending HTTP requests and examining the response. All the tests are written in plain JavaScript and run inside browsers, allowing testers to use the full power of JavaScript and test the problem areas, like browser differences and JS implementation quirks. In addition to the JS libraries there is Python code available that allows automating browser control and makes the results from browsers available to the outside. This makes that HyperTest can easily be integrated into Python test suites; full py.test integration should be available in the near future. Using HyperTest one doesn't need to visit web pages as part of the test cycle anymore: results appear together with 'normal' Python test results. I will be discussing common web application testing problems and provide examples of how to deal with them using HyperTest.
10:10	<b>[49] Introduction to pywinauto (00h30')</b> <i>Speaker: Mr. MC MAHON, Mark</i> To be able to automate GUI testing you first need to be able to automate the GUI. The idea is to operate the GUI much as the user would, and test that the software reacts as expected. Many GUI automation tools require you to record details of the application first - or require you to directly specify the control title to work with. These can be very brittle in the face of software changes. Most commercial tools have their own language - which isn't Python! Pywinauto uses attribute/item access with fuzzy matching to specify controls which could be less brittle than exact matches. It also leaves open the possibility of having localization testing done much more easily than with standard tools. This talk will give a quick introduction to using pywinauto and then show the basics of how its internals work.

## Python Language and Libraries - 40-S2-A01 (03 July 09:00-10:40)

time [id] title

09:00	<b>[33] Introducing PyQt4 for GUI Application Development (00h30')</b> <i>Speaker: Dr. BODDIE, David</i> PyQt4 is a set of bindings for Qt 4, a cross-platform C++ framework used to make graphical user interface (GUI) applications. With the release of PyQt4, Python developers are now able to develop powerful cross-platform applications and deploy them under the GNU General Public License (GPL) or the Qt Commercial License on all platforms that support Qt and Python. We will first briefly discuss PyQt (for Qt 3) and PyKDE (bindings for the K Desktop Environment), and take a look at what has changed in PyQt4. The main part of the presentation will cover the new possibilities that PyQt4 offers developers, including access to Qt's rich text handling features, sophisticated data handling controls, internationalization support, and integration with the Qt Designer GUI design tool.
-------	--

**09:35 [38] Integrating Twisted With Existing Applications (00h30')***Speaker: Mr. ZADKA, Moshe*

Twisted is an event-based framework, with many high-level constructs for handling networking, threads and more. Unfortunately, existing applications are often written so as to own a whole thread of execution. This talk will survey various techniques, using an arranged hypothetical case study, for integrating Twisted with already-existing applications.

**Refereed Papers - 40-SS-D01 (03 July 09:00-09:30)**

time [id] title

**09:00 [16] The Python interpreter as a framework for integrating scientific computing software-components (00h30')***Speaker: Prof. SANNER, Michel*

The focus of the Molecular Simulation Laboratory is to model molecular interactions. In particular we are working on automated docking and molecular visualization. Building and simulating complex molecular systems requires the tight interoperation of a variety of software tools originating from various scientific disciplines. Over the last 10 years we have evolved a strategy for addressing this formidable software engineering problem by developing and integrating software components using the Python language and its interpreter. The basic idea is that the Python interpreter serves as the integration framework and provides a powerful and flexible glue for rapidly compositing components (i.e. Python packages) into applications. We no longer think in terms of programs, but rather in terms of packages, which can be loaded dynamically when needed, and instantly extend our framework (i.e. the Python interpreter) with new capability. We have written more than 25 packages (>1500 classes) providing functionality ranging from scientific visualization and visual programming to molecular simulations. Applications created from these components have been distributed to over 15000 users around the world. In this paper we will describe our approach and various applications, discuss the reasons that make this approach so successful, and present lessons learned and pitfalls to avoid in order to maximize the reusability and interoperability of software components.

**Web Frameworks - 40-S2-B01 (03 July 09:00-10:40)**

time [id] title

**09:00 [25] Working Together on the Web (01h05')***Speaker: Mr. DANGOOR, Kevin*

Python's web ecosystem is richer than ever. For every part of the web application stack, there are more choices than ever, and there are more full-stack web frameworks than ever. Every application has different requirements, which makes choice useful. In my talk, I will discuss how the Python web landscape has been changing and how it's possible to have choices and still avoid reinventing wheels and dividing up our efforts.

**10:10 [104] Developing Applications with the Web Server Gateway Interface (00h30')***Speaker: Mr. GARDNER, James*

The Web Server Gateway Interface is a standard interface between web servers and Python web applications or frameworks, to promote web application portability across a variety of web servers. Although the specification has been around since 2003 there is still a broad lack of understanding in the Python web community about the WSGI and why it is so useful. This hands-on talk will give people new to PEP 333 all the skills they need to start writing their own WSGI applications along with a knowledge of current best practice and a brief overview of the WSGI components that are already available in projects like Paste ([www.pythonpaste.org](http://www.pythonpaste.org)).

**Python in Science - 40-SS-D01 (03 July 09:35-10:40)**

time [id] title

09:35	<p><b>[108] The Enthought Tool Suite for Scientific Applications (00h30')</b></p> <p><i>Speaker: Dr. JONES, eric</i></p> <p>The Enthought tool suite (ETS) is a collection of packages for developing scientific applications. It includes a variety of tools ranging from Envisage, a framework for building scriptable and extensible applications, to Mayavi, a general 3D visualization package. This talk provides an overview of the collection's capabilities and demonstrates their use in a variety of applications. Here is a quick description of the several packages we will cover: Traits ----- Traits forms the foundation of almost every other package in ETS. It allows developers to specify type information for Python class members. This information is used for initializing and validating these members as well as building user interfaces for an object via Traits UI. Additionally, traits supports an observer pattern where "listener" methods are called whenever a member trait changes. Kiva ---- Kiva is a platform independent vector based drawing library for rendering 2D graphics. It is path and affine transform based, and supports alpha transparency. Kiva is used extensively by both the enable and the chaco libraries. Enable ----- Chaco ----- Chaco is Enthought's package for building interactive 2D plots. It architected to support customizable interaction with scientific data as well as efficient rendering of large data sets. Tvtk ---- Tvtk is a "traitsified" layer on the VTK visualization library. In addition to the facilities provided by traits, tvtk provides a more convenient interface as well as more efficient sharing of data between VTK and Python Numeric/NumPy data structures. Mayavi ----- Mayavi is a general purpose 3D visualizaition plugin for Envisage applications developed by Prabhu Ramachandran. This tool is actually 3nd generation of the original Tk based Mayavi application. Mayavi is built on top of Tvtk. Envisage ----- Envisage is a plugin based architecture for developing scriptable and extensible applications. There are a number of plugins that come with ETS including a text editor, python shell, mayavi, and others that can be combined together along with domain specific plugins to develop full features applications. More information on ETS is available at <a href="http://code.enthought.com/ets">http://code.enthought.com/ets</a>.</p>
10:10	<p><b>[64] Envisage - An Extensible Application Framework (00h30')</b></p> <p><i>Speaker: Mr. CHILVERS, Martin</i></p> <p>The Java world has two dominant frameworks for extensible application development: NetBeans and Eclipse. Although many people think of these two projects as IDEs, they are both built upon open architectures that are designed to support generic GUI application development. Treading lightly in the footsteps of these two excellent projects, Envisage attempts to bring similar capabilities to the Python community. The heart of Envisage is its plug-in architecture; in fact, the Envisage core is little more than a system for discovering, loading, starting and stopping plug-ins. Developers construct applications by choosing from the set of supplied plug-ins (possibly the empty set) and then adding their own to provide domain-specific functionality. Although an Envisage application need not have a GUI, Envisage comes with an ever-expanding set of plug-ins that simplify many of the more common and tedious aspects of GUI development (e.g., window management, menubars/toolbars, user preferences). It is worth noting that these 'standard' plug-ins are written in exactly the same way as those written by an application developer. Each plug-in uses the Envisage extension mechanism not only to contribute to what came before it, but also to define how future plug-ins can contribute to it. This mechanism is simple but consistent, and ensures that even extensions to an application are extensible. This presentation will introduce the Envisage plug-in architecture, and demonstrate how applications both with and without GUIs can be built with it.</p>

## **Agile Development - 40-SS-C01 (03 July 11:00-12:40)**

time [id] title

11:00	<p><b>[115] Literate testing with doctests (00h30')</b></p> <p><i>Speaker: Mr. GEDMINAS, Marius</i></p> <p>Doctests are one way of writing automated tests in Python. Doctests can be used for many purposes. I will demonstrate (with examples) the use of doctests for documentation, design, unit testing, and functional/acceptance testing. I will show some interesting features of the doctest module such as pattern matching or diffing. I will show how to collect and aggregate doctests into unittest-compatible test suites. I will talk about certain caveats and demonstrate some useful techniques for writing the tests themselves. About the author: Marius has been writing Python code for the last 4.5 years non-stop. He is deeply convinced about the gains of automated tests, and thinks that if you do not already write doctests for your Python code, then you should start doing so.</p>
11:35	<p><b>[5] zope-cookbook.org, an open Zope book based on agile documentation principles (00h30')</b></p> <p><i>Speaker: Mr. ZIADÉ, Tarek</i></p> <p>zope-cookbook.org is an open book project that contains recipes and tutorials for Zope 3 developers. This book is written in several languages (French and English so far) and based on agile documentation principles. It uses Python doctests to provide participants the ability to write their recipes like they would write code, by using a test-driven approach. The code examples can therefore reach a high level of quality, since they are executed for real, and evolves with the framework in use. After a brief technical overview on how the project runs, and some extracts, the presentation will explain how a book project can be driven exactly like a software one, and why it is very important in such a dynamic and moving project like Zope 3. EuroPython would be a nice place to promote the project to the Zope and Python community, to get direct feedback and even maybe new writers!</p>

**12:10 [10] Planning your tests with morphological analysis (00h30')***Speaker: Mr. THEUNE, Christian*

Morphological analysis is a powerful approach to generic problem solving. Using this method you can reliably plan for the minimal amount of test cases you should cover for your application. This talk will demonstrate how to apply the methodology and how to manage test plans with a test planning software based on morphological analysis.

**Python Language and Libraries - 40-S2-A01 (03 July 11:00-12:40)**

time [id] title

**11:00 [0] Python and PostgreSQL - a match made in heaven (00h30')***Speaker: Mr. MASSA, Harald Armin*

PostgreSQL is a heavenly match to Python. Both share the same kind of licence, both have an excellent community, both work perfectly together. I will destroy bad PostgreSQL myths, give an overview of using PostgreSQL to interface with Python, and as a special feature show how to use Python inside PostgreSQL to write stored procedures. The business aspect, the possibilities of rollout and also the availability on the main platforms will be discussed.

**11:35 [122] Index & Search: itools.catalog (00h30')***Speaker: Mr. BELMAR-LETELIER, Luis*

There are not many solutions for indexing and searching in Python. Among them itools.catalog is the only one 100% Python and actively maintained. This talk will explain the usage of itools.catalog. We will also define its functional scope, this is to say, what it can do and what it cannot do. And we will put it in context with other solutions available to Python.

**12:10 [72] Using CSV as an indexed persistent layer (00h30')***Speakers: Mr. CAUWELIER, Hervé, Mr. DERAM, Nicolas*

This talk will introduce how itools.csv provides a high-level API to handle CSV files. From the loading of a simple CSV file to the serialisation of a complex object, we will demonstrate automatic type marshalling, rows handling, and the search engine. We will address underlying concepts, such as datatypes encoding and decoding, indexing, and querying by using a simple syntax. Extending the API will also be illustrated. Change your point of view on CSV!

**Python in Science - 40-SS-D01 (03 July 11:00-12:40)**

time [id] title

**11:00 [80] Introduction to Code\_Aster (00h30')***Speaker: Mr. AUBRY, ludovic*

Code\_Aster is a simulation tool for mechanics developed over the past twenty years at Electricité de France Research and Development. In the year 2000, its home-brewed scripting language was replaced with Python. This talk will present the features of Code\_Aster and the benefits gained by the use of Python.

**11:35 [98] OpenAlea - A platform for plant modelling, analysis and simulation (00h30')***Speaker: Mr. PRADAL, Christophe*

The aim of the plant architecture research community is to understand the biological processes involved in the function and growth of plants with explicit representation of their topology and geometry. To understand these systems, which may be quite complex, researchers in botany, ecophysiology, forestry, horticulture, applied mathematics and computer science share experimental data, plant models and software tools. Research in this domain concerns several different scales, such as plant cells, plant organs, entire plants and plant populations. To support this research, we are developing OpenAlea, an open source software platform implemented in Python. OpenAlea is used for the integration and comparison of diverse models and tools provided by the research community. We use Python to glue together components implemented in various languages (e.g. C, C++ and Fortran). Standard wrapping tools, such as Boost.Python, Swig and f2py, are used to support the integration process. Different components can use common data structures such as sequences, tree graphs and multiscale tree graphs. To ease the communication between components, a unique interface has been specified for each data structure. The Zope component framework is used to define interfaces and adapters. Three types of components are integrated into OpenAlea: - analysis of plant architecture (e.g. AMAPmod), - geometric representation and visualization of plants at different scales (e.g. PlantGL), - and simulation models of ecophysiological processes (e.g. RATP, Archimede, PyCaribu, etc.). Users can combine components from OpenAlea and other Python scientific libraries into customised work flows according to their specific needs. Training courses on Python and OpenAlea's modules are given in France and other countries (e.g. Thailand, Brazil, etc.) or through an e-learning platform.

**12:10 [109] Factory monitoring with Pylons, XML-RPC and SVG (00h30')***Speaker: Dr. COLLINS, Rob*

I write Python applications for a company specialising in industrial factory monitoring and scheduling. Most recently we have developed a web application for remote monitoring, sending text message alarms when machine hoppers run low. We're using the Pylons web framework, Myghty templating, XML-RPC and SimPy simulation. The dynamic graphical display of the factory is written in SVG, for Firefox (v1.5) web browser. We will look at interesting issues such as \* the choice of framework and templating system, \* the importance of decoupling threads using a log file and XML-RPC, \* historical correction of data, \* and why we needed simulation.

**Web Frameworks - 40-S2-B01 (03 July 11:00-12:40)**

time [id] title

**11:00 [26] The Django Web Framework (00h30')***Speaker: Mr. WILLISON, Simon*

Django is a Python web framework designed to enable "web development with journalism deadlines". This talk will provide an overview of the framework, discuss the environment in which it was developed and describe the evolution of the framework in the year since it was first released.

**11:35 [12] MoinMoin Wiki Development (00h30')***Speakers: Mr. WALDMANN, Thomas, Mr. SCHREMMER, Alexander*

MoinMoin is a popular and powerful wiki engine in Python. The talk will give an introduction to the MoinMoin core source code, extension concepts and extension development. Shortly, we will present the flexibility of MoinMoin and how the user base (Python, Apache, Ubuntu, ...) looks like. We will give an architecture overview as well as explain some code and show how you can write your own.

**12:10 [52] Generating content types and workflow with ArchGenXML (00h30')***Speaker: Mr. REES, Reinout van*

For many people, ArchGenXML is the most attractive way to get started with Plone development. Generating your content types from a UML class diagram is easy and fast, especially as ArchGenXML sets up all the "bookkeeping code" for you: the `__init__`, the `Install.py`, etc. The presentation will cover this and I'll also show how to modify the generated code to your special needs. ArchGenXML is great at generating a complete workflow out of UML state diagrams. You get good code that hardly ever needs modification from a UML diagram that you can readily show to your customer - and he'll be able to understand it (mostly). I'll show how to do this in the presentation. Depending on the improvements made on ArchGenXML I hope to show our support for zope3 goodies in plone 2.5.

**Agile Development - 40-SS-C01 (03 July 14:00-15:40)**

time [id] title

**14:00 [54] eXtremeManagement of projects with Plone (00h30')***Speaker: Mr. VAN REES, Maurits*

eXtremeManagement is a Plone product that helps in developing using the Extreme Programming methodology. It helps steer a project to success by a focus on planning, on estimating and on keeping a log of your hours worked. My colleagues and clients at Zest Software like it. Goals of the talk: - Get a feel for the planning and estimating parts of XP. - Demonstrate the eXtremeManagement product and show how it helps with this. - Get improvement ideas for eXtremeManagement and probably for your own project management software.

**14:35 [75] Distributed Source Code Management tools (00h30')***Speakers: Mr. BELMAR-LETELIER, Luis, Mr. CAUWELIER, Hervé*

A new breed of Source Code Management tools has exploded in the Free Software world. These are the Distributed SCMs, an indispensable tool for the agile developer. This talk will explain how a Distributed SCM is different from a Centralized one, and why this makes a special difference in an agile environment. We will expose our working experience at Itaapy with these tools: - how they became the central part of our production process. - what was the motivation to move from CVS to GNU Arch, and then from GNU Arch to Git and Cogito. We will close with a tour through the most prominent solutions available: Bazaar-NG, Darcs, GIT and Mercurial. And we will explain why we chose GIT.



**15:10 [62] Source code management for a distributed team. (00h30')***Speakers: Mr. ALEXANDER, Steve, Mr. COLLINS, Robert*

One way to manage the source code flow for medium size projects with distributed teams. Canonical manages the source code for its Launchpad project using bzr. Launchpad has 15 developers spread over 5 continents who primarily work from home. We demonstrate how a feature goes from being just an idea through development, code review and finally being committed to the official source code tree after it passes our regression test suite.

**Python Language and Libraries - 40-S2-A01 (03 July 14:00-15:40)**

time [id] title

**14:00 [68] Programming Avalon with IronPython (00h30')***Speaker: Mr. QUINLAN, Brian*

This talk will use a simple (but useful) example application to demonstrate how Windows Presentation Foundation (WPF) applications can be build using Python. The talk will begin by explaining key Microsoft .NET concepts and how they can be accessed though IronPython. Authoring simple eXtensible Application Markup Language (XAML) interfaces will also be demonstrated, both using Microsoft's GUI tools and by writing XAML code directly. Finally, the key points in building and packaging a complete WPF application will be illustrated.

**14:35 [83] Design by Contract in Python: Present and Future (00h30')***Speaker: Mr. BINGHAM, Aaron*

Design by Contract (DbC), as proposed by Bertrand Meyer, is an approach to software development whereby the expectations (preconditions) and guarantees (postconditions and class invariants) of methods are formally specified. These formally-specified contracts can then be checked automatically at runtime. First, Design by Contract as proposed by Meyer is defined, its strengths and limitations are described, and the rationale behind the various aspects of the definition are explained. Second, several implementations of DbC for Python, each taking a different approach, are introduced. The implementations are compared to one another and to Meyer's proposal, with particular attention to their suitability for production use. (At least one implementation will be demonstrated live.) Third, attributes desired of a future production-quality DbC system for Python are outlined, and possible evolutionary paths based on existing implementations are proposed.

**Python in Science - 40-SS-D01 (03 July 14:00-15:40)**

time [id] title

**14:00 [46] MDP 2.0 - A data processing framework for scientific development and education (00h30')***Speaker: ZITO, Tiziano*

We present release 2.0 of the Modular toolkit for Data Processing (MDP), a data processing framework written in Python and based on numpy (the most popular numerical extensions to Python). From the user's perspective, MDP consists of a collection of trainable algorithms or other data processing units (nodes) that can be combined into data processing flows. Given a sequence of input data, MDP takes care of successively training or executing all nodes in the flow. This structure allows to specify complex algorithms as a sequence of simpler data processing steps in a natural way. Training can be performed using small chunks of input data, so that the use of very large data sets becomes possible while reducing the memory requirements. Memory usage can also be minimized by defining the internals of the nodes to be single precision. From the developer's perspective, MDP is a framework to make the implementation of algorithms easier. The basic class 'Node' takes care of tedious tasks like type and dimension checking, leaving the developer free to concentrate on the implementation of the training and execution phases. The node then automatically integrates with the rest of the library and can be used in a flow together with other nodes. MDP 2.0 introduces some important structural changes. It is now possible to implement nodes with multiple training phases and even nodes with an undetermined number of phases. This allows for example the implementation of algorithms that need to collect some statistics on the whole input before proceeding with the actual training, or others that need to iterate over a training phase until a convergence criterion is satisfied. The ability to train each phase using chunks of input data is maintained if the chunks are generated with iterators. Moreover, it is now possible to define nodes that require supervised training in a very straightforward way by passing additional arguments (e.g., labels or a target output) to the 'train' method. Moreover, new algorithms have been added, expanding the base of readily available basic data processing elements. Currently implemented algorithms include Principal Component Analysis, two flavors of Independent Component Analysis, Slow Feature Analysis, Gaussian Classifiers, Growing Neural Gas, Fisher Discriminant Analysis, and Factor Analysis. MDP has been written in the context of theoretical research in neuroscience, but it has been designed to be helpful in any context where trainable data processing algorithms are used. Its simplicity on the user side together with the reusability of the implemented nodes make it also a valid educational tool. As its user base is steadily increasing, MDP appears as a good candidate for becoming a common repository of user-supplied, freely available, Python implemented data processing algorithms. <http://mdp-toolkit.sourceforge.net>

14:35	<p><b>[55] Visualizing behavior of ambient sensor networks (00h30')</b></p> <p><i>Speaker: DE RIDDER, Theo</i></p> <p>Ambient sensor networks represent distributed systems that challenge the illusion that Moore's law will always facilitate more complex software in the near future. The need for extreme low power consumption in myriads of miniscule sensors is a driving force to reduce software again to its basics. It appears that good old concepts enable new animals like hopping and gossiping nano-agents. But a lot of validating experiments are yet to be done to understand and control behavior of sensor clouds in realistic ambient environments. This talk is about a framework to simulate and monitor dynamic sensor-networks in 3D, combining the capabilities of Blender, numpy and Python. Blender is used for modeling and animating 3D worlds. Cluster behavior is represented in numpy as scalable matrix algebra similar to Matlab. Python is the universal and transparent glue for any component.</p>
15:10	<p><b>[113] Using Python in the Development of a Grid User Interface for Distributed Data (00h30')</b></p> <p><i>Speaker: SOROKO, Alexander</i></p> <p>Details are presented of GANGA, the Grid user interface being developed to enable large-scale distributed data analysis within High Energy Physics. GANGA helps users to configure, execute, monitor and manage their large scale computing tasks. We show how the basic functionality implemented in a small GANGA core can be enhanced by tailored plugins provided by specific user communities. We show how Python metaclasses and descriptors are used to build user-friendly interfaces to the plugged modules in a dynamic way. The interface allows simultaneous use of an interactive prompt based on IPython, full scripting capabilities as well as a dynamically created GUI based on PyQt. We describe the optimisations developed for the serialization of GANGA objects and finally, we present GANGA approaches to the configuration and help systems.</p>

## **Web Frameworks - 40-S2-B01 (03 July 14:00-15:40)**

time [id] title

14:00	<p><b>[34] An Introduction to TurboGears (00h30')</b></p> <p><i>Speaker: Mr. DANGOOR, Kevin</i></p> <p>Since its release in September, TurboGears has quickly grown a large and active community of developers. With TurboGears 1.0 currently in testing, there is quite a bit to show off that hasn't been presented before. In the "20 Minute Wiki", I show the end-to-end creation of a simple application. In this presentation, I will present a feature tour (with code snippets) to highlight some of the high-level facilities TurboGears offers.</p>
14:35	<p><b>[105] Pylons: A Modern Python Web Framework (00h30')</b></p> <p><i>Speaker: Mr. GARDNER, James</i></p> <p>Pylons is a lightweight web framework emphasising flexibility and rapid development. Pylons combines the very best ideas from the worlds of Ruby, Python and Perl, providing a structured but extremely flexible Python web framework. It's also one of the first projects to fully leverage the Web Server Gateway Interface standard which facilitates extensive re-use and flexibility — but only if you need it. Out of the box, Pylons uses best practice principles to make web development fast, flexible and easy. (<a href="http://www.pylonshq.com">www.pylonshq.com</a>) This talk will take the audience on a whistle-stop tour of Pylons, from installing the software to deploying a finished application.</p>
15:10	<p><b>[22] Bebop: A Zope3-based Groupware (00h30')</b></p> <p><i>Speaker: Dr. OESTERMEIER, Uwe</i></p> <p>Bebop is an open source (GPL) groupware that tries to combine collaboration support and content management with a user-friendly interface. Group aspects (e.g. who is online, who works on what, versioning conflicts, etc.) are seamlessly integrated into the user's local workspace. Bebop comes in two forms: a web client for casual users, and a standalone client for users with the need for professional editing tools. The web client provides wiki, blog, news, and repository views on Zope3 standard content objects. As an Ajax application it offers live search, smart wiki links with various edit options, and WYSIWYG editing and merging. A two-way event notification (cf. Nevow's LivePages) for Zope3 is under development. Although the Ajax client feels in many respects like a standalone application, it still suffers from a limited integration with the desktop and editing tools. To overcome this major shortcoming of web-based applications a standalone client written in wxPython will be the major focus of the Bebop project. This client stores all content in the filesystem and shows in real time who's online, who's sharing the same point of view, and visualizes all modifications and versioning conflicts. The paper will focus on Bebop's architecture and sketch future developments. Bebop: <a href="http://svn.kmrc.de/LivePage: http://svn.zope.org/*checkout*/zope3org/trunk/src/zorg/live/README.txt">http://svn.kmrc.de LivePage: http://svn.zope.org/*checkout*/zope3org/trunk/src/zorg/live/README.txt</a></p>

## **Games - 40-S2-B01 (03 July 16:00-17:40)**

time [id] title

16:00	<b>[47] Pre-Mapping a GSM Network Environment for a Pervasive Game using Python for Series 60 Phones (00h30')</b> <i>Speaker: Mr. OPPERMANN, Leif</i> Pervasive Games are a new type of game that takes place outside in the wild and makes sense of the user's context, often based on her position. Mobile phones are the perfect ubiquitous device for this type of game but usually come without a GPS device which could be used for positioning. Operator based positioning can deliver a GPS-compatible coordinate in latitude and longitude (WGS84), but this involves a fee for every position look-up and is only available on a few networks. Client based positioning based on the GSM network's cell IDs is an alternative solution, where the phone's current serving cell is utilized to infer the user's location either in an abstract graph space or in WGS84 coordinates by using look-up tables of previously seen and geocoded cell IDs. This talk covers how Python for Nokia Series 60 phones has helped us to collect the data for the client based positioning approach and how we use it to prepare for a location-based multiplayer game across the 3 cities of Nottingham, Derby and Leicester which will go online on Valentines Day 2007. The game is going to run on a number of mobile phones. We are using Python for rapid prototyping of required tools and potentially also the gaming interface. Parts of the software could potentially be made available to the community. Parts of the software have been developed within a 6th Framework EU funded integrated project on pervasive gaming (IPerG) in which the author and his affiliation participate. Links: <a href="http://www.pervasive-gaming.org/">http://www.pervasive-gaming.org/</a> <a href="http://makinglovecity.blogspot.com/">http://makinglovecity.blogspot.com/</a> <a href="http://www.lovecity.tv/">http://www.lovecity.tv/</a>
16:35	<b>[48] Bub-n-bros = Bubble Bobble + 10 players + general craziness from Python's flexibly (00h30')</b> <i>Speaker: Mr. RIGO, Armin</i> Bub-n-bros is an Internet version of the original Bubble Bobble game: everyone play a dragon that fires bubbles with which he catches monsters. This version has a fast-paced crazy-bonuses feeling to it that is the true landmark of not only ten dragons jumping all around but a really flexible programming language that does not restrict inventiveness.
17:10	<b>[78] LLG Python: Geometric Computing in Python for next generation games and entertainment (00h30')</b> <i>Speaker: Mr. FARRET, Jérémie</i> Next generation games and entertainment raise new questions in terms of architecture, design and development. Parallel processing hardware architectures enable disruptive approaches in terms of real time simulation and visualization. Storage capabilities grow ten folds, along with content creation and hyperlarge scenes crucial problems. Massively multiplayer online gaming brings brand new perspectives on networking applications, and concerns about participative content creation and the corresponding bandwidth issues. Finally, 3D makes its way to embedded, mobile devices, spawning with a number of specific difficulties. LLG Python proposes an original, Python based environment unifying volumic data representation and processing for real time applications. This unification through 3D programming and scripting enables defining solutions to many next generation gaming and entertainment upcoming challenges.

**Python Language and Libraries - 40-S2-A01 (03 July 16:00-17:40)**

time [id] title

16:00	<b>[4] Designing Unicode-aware Applications in Python (00h30')</b> <i>Speaker: Mr. LEMBURG, Marc-André</i> The talk presents ways of designing Python applications to be Unicode-aware from ground up. I've held the talk at the LSM 2005 conference in Dijon, France and people were very interested. You can download the PDF for a preview: <a href="http://www.egenix.com/files/python/LSM2005-Developing-Unicode-aware-applications-in-Python.pdf">http://www.egenix.com/files/python/LSM2005-Developing-Unicode-aware-applications-in-Python.pdf</a>
16:35	<b>[7] Internationalization in Python with PyICU - Chandler, a case study (00h30')</b> <i>Speaker: Mr. BOSSUT, Philippe</i> As a cross platform desktop client, Chandler needed a rich and robust internationalization (i18n) strategy. Alas, though Python does have some good localization (l10n) support, it does not provide a robust i18n strategy covering things like: localizable searching, sorting, currency and date formatting and boundary recognition. The Chandler project decided to bite the bullet and bring the ICU library into the Python fold providing a PyICU API. Though using PyICU, Chandler also uses the gettext model and .po format people in the Open Source community use and love for translation. The talk will cover the various aspects of the internationalization and localization challenges with code samples from Chandler and demonstration.
17:10	<b>[112] Common mistakes you can make using pytz and datetime (00h30')</b> <i>Speaker: MIKALAJŪNAS, Ignas</i> With pytz and datetime being included in the python2.4 distribution most people don't expect to have any trouble with using them. While they seem easy on the surface both of these libraries have a few gotchas that are very easy to encounter unless you really know them both. Working on a calendaring heavy web applications (SchoolTool/SchoolBell) I have encountered a lot of datetime related bugs myself. So I will try to demonstrate common mistakes made by developers and explain the right way of performing operations that are otherwise difficult to get right.

**Python in Science - 40-SS-D01 (03 July 16:00-17:40)**

time [id] title

**16:00 [116] Taming the beast: Using Python to Control ATLAS Software (00h30')***Speaker: QUARRIE, David*

ATLAS, one of the four experiments at the CERN LHC accelerator, uses the Athena/Gaudi component architecture to build and run its software applications. Python was initially introduced as a configuration language for the hundreds of C++ components, but soon, thanks to the development of a powerful C++ binding layer, it started pervading all application domains from job management, to application control to interactive analysis. In this talk we will survey both successes and challenges brought by the introduction of Python and we will discuss our ongoing efforts to use the flexibility of Python to make easier and safer for ATLAS physicists to build, configure and run their applications.

**16:35 [111] Interfacing Python and the C++ Frameworks used by High Energy Physicists (00h30')***Speaker: Dr. MATO, Pere*

In this paper we present the developments in the area of interfacing Python with large-scale C++ frameworks, driven by the needs of the new generation of High Energy Physics experiments currently in preparation at CERN. The physics software being developed for detector simulation, data reconstruction, and data analysis is mainly written in C++. Yet, scripting is an essential functionality of the complete software system that we are providing to the physicists, and therefore it is essential to have a performing, easy to code to, and flexible interface from C++ to Python. Python is being used for application configuration, rapid prototyping, and interactive physics analysis. The Python interface has been built on top of the extended C++ reflection capabilities provided by such tools as CINT and Reflex. This reflection information, in terms of dynamic loadable libraries, can be used for the creation of Python bindings to C++ classes, functions, variables, etc., in a fully automated manner, since this information is already provided for by many end-user libraries for other purposes, such as object persistency. A consistent and automatic mapping of any C++ construct into an equi-purpose Python construct is achieved easily and elegantly. We finalize with a few use case examples demonstrating the capabilities of this interface.

**17:10 [94] Finding Needles in a Huge DataStack (00h30')***Speaker: Mr. FRANCESC, Altet*

Many tools exist in the Python world to handle persistent data. Most of them are high-level wrappers to access well-known relational databases (Oracle, Postgres, MySQL...), while others are wrappers to highly-efficient, specific-purpose libraries (bsddb, NetCDF3...). Others have developed their own specific formats to fulfill their own requirements. In the data-hungry world of scientific computing, one usually prefers (with good reason) solutions that are not only fast but also well-tested and, perhaps more importantly, have outstanding backward and forward format compatibility. Scientific applications also tend to focus on the most efficient ways to find the "needles in the haystack" of massive amounts of data. We will begin the talk with a description of HDF5 [1], an emerging standard format to store scientific and other data. Its main features will be covered, and the contexts where it can be applied to an advantage will be discussed. We will then introduce PyTables [2], a well-known and widely adopted solution implemented in Python for manipulating potentially huge HDF5 datafiles easily and efficiently. We also plan to offer a sneak preview of the next-generation PyTables toolkit, with its greatly improved indexing and search capabilities. The PyTables discussion will include benchmarks of the latest versions, to give an idea of its lookup speed and performance as compared to other well-established standard databases and toolkits. Users will learn what they can expect from the next-generation PyTables and how it can help them to find specific data (the needle) in huge (terabytes and petabytes) datasets very rapidly. In conclusion, we will unfold our master plan for the future domination of the world by PyTables and its growing family. [1] <http://hdf.ncsa.uiuc.edu/HDF5/> [2] <http://www.pytables.org/>

**Keynote - Main Auditorium (03 July 18:00-19:00)**

time [id] title

**18:00 [30] Children First! (01h00')***Speaker: Dr. KAY, Alan*

Why and how the worldwide programming community (and especially the Python community) should get interested in children's education, especially in the 3rd world, and help by making children's learning environments. I will show such an environment that I'd like to see done in Python (because I think it will spread better and be a maintained better around the world in Python).

## Tuesday 04 July 2006

### **Business and Applications - 40-SS-C01 (04 July 09:00-10:40)**

time [id] title

09:00 **[29] Achieving High Performance In Mercurial (00h30')**

*Speaker: O'SULLIVAN, Bryan*

Mercurial is a distributed revision control system designed for performance and scalability. Although written in Python, it is faster than most other revision control tools. This paper discusses the techniques used in Mercurial to achieve such high performance, and compares its performance on common operations with three other revision control systems: CVS, Subversion, and Perforce. It also introduces some novel software development techniques that are made possible by Mercurial's high performance.

09:35 **[37] Developing MailManager (00h30')**

*Speaker: Mr. KEVIN, Campbell*

MailManager is an email response management product written in Python. It is built on top of Zope and runs against an SQL database. The source code is freely available under the GPL. MailManager is sold to customers either as a web based application, an appliance server, or via support contracts. Additionally, customers often pay for modifications of the software, which are normally merged into the main development release. Some of the main challenges for development include the additional overheads of maintaining datasets for multiple revisions of the product for open source users, and ensuring compatability with a large range of target platforms. An international customer base requires i18n support throughout the product. Additionally, continual feature requests need to be developed and managed against the public release cycle. The development process makes heavy use of a test based methodology. BuildBot is used to run the test suite automatically on all supported platforms. Some of this code is also reused in order to provide extensive runtime monitoring of the product, and ensure it is functioning correctly. Logicalware (the company behind MailManager) was one of the first companies in the UK to receive VC funding for developing an open source product.

10:10 **[42] Snaking the Web (00h30')**

*Speaker: Mr. FRANZ, Markus*

Two research projects of NEC and the University of Hannover could not be realized because of technology faults. With Python a 18 year old student from Germany solved the problems and created a new-class metasearch engine for the WWW. The development process with problems like high-volumes of data, web crawling, performances iusses and other topics was easy with Python like with no other programming language.

### **Python Language and Libraries - 40-S2-A01 (04 July 09:00-10:40)**

time [id] title

09:00 **[35] Python as a domain specific language (00h30')**

*Speaker: Mr. HAMMARQUIST, Anders*

By using metaclasses and other powerful features of Python it is possible to use Python as a domain specific language. We will show how we replaced a domain specific language in CAPS with Python, and discuss our reasons for doing so. Using metaclasses and other meta-features, it is possible to greatly alter the behaviour of Python. We have exploited these to give us what we needed for our domain specific language, which is used to describe database object and to code fuctionality for them, while still retaining a clear syntax. The result is something not-quote-Python but still clear if you know what we are trying to do.

09:35 **[32] The Chandler Repository, a Database for Python Objects (00h30')**

*Speaker: VAJDA, Andi*

OSAF's flagship project, Chandler (<http://chandler.osafoundation.org>), is a personal information manager (PIM). As such, it needs to persist much of a user's personal data. Three years ago, OSAF embarked on developing an embedded desktop database for persisting the kinds of Python objects used in a PIM. Along with its data model, the Chandler Repository is the foundation layer in the Chandler software stack. In this talk, we would like to present some of the design decisions we made early on and discuss how they have served us over time. In particular, we'd like to address the topics of: - the language neutral data model - managing references between objects with natural indexes - handling concurrency without locks by merging changes - the pythonic schema API - queries via collections - caching via collection indexes and notifications



**10:10 [97] zc.buildout, an automated application assembly tool (00h30')***Speaker: Mr. FULTON, James*

Applications and application-development environments are often built from parts, such as Python packages, databases, libraries, and configuration. The assembly process can be quite involved. It is best to use tools to automate the assembly process. Setuptools is a project that provides a packaging system for Python. It extends Distutils with the ability to define package dependencies and automated package installation. It provides automated platform-appropriate script generation. It allows multiple versions of packages to be installed, such that different applications can use different package versions as needed. Setuptools is targeted toward installing packages into a Python installation. It is not well suited to installing packages into a development environment. Setuptools assembles applications at run time, leading to undesirable non-determinism when new packages are installed. We have developed an application and framework, zc.buildout, that provides automated application assembly leveraging setuptools. It can be used to assemble any application, but it is especially well suited to and is designed for assembling Python applications. Applications are defined with configuration files. Configuration files specify a collection of parts, along with their configuration data, making up the application. Parts are implemented by recipes. Recipes are defined as setuptools packages. Recipes are provided for installing Python packages and scripts leveraging setuptools' automated installation facilities, providing an improved mechanism for leveraging setuptools in development environments and other situations where installing into a Python installation isn't appropriate. The buildout system provides install-time, rather than run-time application assembly, providing greater predictability.

**Refereed Papers - 40-SS-D01 (04 July 09:00-10:40)**

time [id] title

**09:00 [18] An Introduction to Test Driven Code Generation (00h30')***Speaker: Dr. MARVIE, Raphael*

Agile Software Development promotes the use of techniques such as Test Driven Development (TDD) and Automation in order to improve software quality and to reduce development time. Code generation represents a way to achieve automation, reducing repetitive and error-prone tasks. Code generation is well accepted, writing a code generator is not necessary that hard, however it is not trivial to decide when and how to embrace code generation. Moreover, it is even harder to embrace at the same time code generation and TDD, wondering for example "How to test drive generated code?" or "How to build a generator following a test driven approach?" This paper aims at providing hints to answer those questions. It presents an iterative approach named Test Driven Code Generation. The main principle is to gain knowledge about the application during the first iterations of its development process and then to identify how to implement code generation. As code generation should not drive you out of TDD, we provide hints to marry both approaches in order to empower your developments. Using a simple 3-tier web application, this paper is illustrated with the Python standard unittest module, CherryPy web application server and Cheetah templating system.

**09:35 [20] Developing Ambient Displays for Collaborative Work (00h30')***Speaker: Dr. MCCALL, Rod*

This paper discusses a prototype system known as the Ambient Workplace (AW) that lets people monitor work patterns within a group of co-workers. The AW provides information in the user's peripheral attention zone with the objective being that the user soaks up the information without needing to specifically attend to it. Information is displayed within standard applications such as the Windows Active Desktop, Outlook or Word. At present the system displays various types of awareness information, for example the number of interactions a specific user or group of users has initiated and whether they are online. It also displays global information about the volume and type of work for the entire group. The Ambient Workplace uses Python and displays information using Scalable Vector Graphics (SVG). It uses a client/server architecture built upon the XMPP instant messaging protocol and retrieves information from a MySQL database. The objective was to build a system which can be easily ported to other platforms such as mobile phones and smart boards as well as to leverage existing libraries while allowing rapid prototyping of the user interface.

**10:10 [36] PyJIT: dynamic code generation from runtime data (00h30')***Speaker: Mr. BURTON, Simon*

PyJIT is a framework for providing just-in-time compilation from within Python programs. The idea is to generate machine code at runtime, and inline our data as we go. This can provide significant speed-ups versus compiled code, by reducing memory bandwidth use and simplifying the executable code. At the core we use the Low Level Virtual Machine (LLVM) which has facilities for creating machine dependant code on the fly. PyJIT handles conversion to LLVM Static Single Assignment form, and has a compiler that accepts a subset of the Python syntax, including basic control flow. We demonstrate applications in: vectorized operations and numerical linear algebra, tree structures for machine learning, and optimization techniques using interval arithmetic.

**Web Frameworks - 40-S2-B01 (04 July 09:00-10:40)**

time [id] title

09:00	<b>[74] i18n made easy, illustration with CherryPy (00h30')</b> <i>Speakers: Mr. DERAM, Nicolas, Mr. CAUWELIER, Hervé</i> After a quick introduction to i18n (internationalization), we will show how it is fast and efficient to turn an application into a multilingual application using the itools library. The powerful framework CherryPy will be used to highlight in a fully functional example, in particular the four following aspects: * automatic message extraction in Python code; * automatic message extraction in XML templates; * speed improvement by using one template per language; * language negotiation API. Those aspects allow the internationalization of a Python application, even outside the Web context.
09:35	<b>[28] CPSBayes, a multipurpose naive bayesian classifier for Zope (00h30')</b> <i>Speaker: Mr. ZIADé, Tarek</i> Naive Bayesian classification is widely used in spam detection systems, like SpamAssassin. CPSBayes is a Zope 3 / Five product that provides a naive Bayesian classifier tool over a CMF portal, with a SQL backend storage. This talk presents the tool and its primary use case: classifying incoming emails in themes, before they become web documents sent in specific workflows. Moreover, the presentation focuses on other uses of the tool for other tasks over a CMS (Content Management System) or an ECM (Enterprise Content Management) like automatic filling of metadata. This kind of automated task enhances some features that often get less accurate when the document corpus grows because users never do those tasks (search, etc.)
10:10	<b>[90] Version content in Plone with CMFEditions (00h30')</b> <i>Speaker: Mr. WEBER, Grégoire</i> With CMFEditions Plone content may be versioned. Around the conference a 1.0rc or 1.0final will be available. It works with all standard Plone content out of the box. It offers strong architectural support for future use cases. There are already a handful of sites productive with CMFEditions alpha versions.

## **Business and Applications - 40-SS-C01 (04 July 11:00-12:40)**

time [id] title

11:00	<b>[60] LiSoG - Linux Solution Group. Connect real users, technology, applications and businesses. (00h30')</b> <i>Speaker: Mr. MACHE, Niels</i> The Linux Solution Group ( <a href="http://www.lisog.org">www.lisog.org</a> ) is a cooperation platform for companies and authorities with a strong interest in doing business or running applications based on Linux and Open Source. Members of the LiSoG range from global enterprises like IBM, Novell and Red Hat to small businesses, authorities and users. The session will explain the goals and targets of the LiSoG solution group. The speaker Niels Mache is a board member of the Linux Solution Group.
11:35	<b>[71] ZEA Partner network explained (00h30')</b> <i>Speaker: Mr. HEYMANS, Xavier</i> The partnership of Zea ( <a href="http://www.zeapartners.org">www.zeapartners.org</a> ) is focused on growing open source and open source business by pooling resources for collective action as promotion and projects. With 20 partner companies in 11 countries, ZEA provides a broad reach of expertise in content management. The ZEA network currently totals a combined staff of over 150 employees and a turnover in excess of € 9 million. The talk will investigate the challenges facing network of Sme's, the strategy and actions of Zea Partners. The speaker Xavier Heymans is CEO of Zea Partners.
12:10	<b>[82] Introducing python into industrial environment applications (00h30')</b> <i>Speaker: PLIGER, Fabio</i> Two years have passed since the company I work for (SIA) decided to use Python to develop applications. Most of those applications are for pharmaceutical companies with high industrial and production needs. So, I would like to talk about why we have chosen python, about the main problems using Python with industrial SCADA frameworks and also about some results and tips. To wrap it up, I would like to comment on what we think the Python evolution within our company will be.

## **Python Language and Libraries - 40-S2-A01 (04 July 11:00-12:40)**

time [id] title

11:00	<p><b>[81] An Introduction to PyPy (00h30')</b></p> <p><i>Speaker: Mr. HUDSON, Michael</i></p> <p>PyPy is an implementation of Python written in Python itself. It aims at performance and flexibility. This talk will introduce the project and attempt to explain why it is interesting and useful. The intended audience is people who have heard of PyPy, have a vague idea that it is interesting but are not really sure why, though if you've never heard of PyPy I'll be delighted to explain things to you too :)</p>
11:35	<p><b>[93] PyPy architecture session (00h30')</b></p> <p><i>Speakers: KREKEL, Holger, PEDRONI, Samuele</i></p> <p>In this session we will present and interactively discuss with the audience the basic architectural picture of PyPy - a next generation Python compiler and interpreter. We will emphasize various emerging possibilities enabled by our architecture and ask the audience for questions. In particular, we'll describe the following architectural pieces and point out extension and optimization possibilities: - Language Implementation: Bytecode Interpreter and Object Space interaction - Translation to low level languages (e.g. C/LLVM) - Translation to higher level languages (e.g. .NET/Squeak) - JIT-compiler architecture - Interweaving of Garbage Collection, threading and stackless operations into the translation process</p>
12:10	<p><b>[107] What can PyPy do for you? (00h30')</b></p> <p><i>Speakers: Mr. BOLZ, Carl Friedrich, Mr. RIGO, Armin, Mr. VAN RIET PAAP, Eric</i></p> <p>This talk describes novel features of the PyPy project (a next generation Python interpreter) which are made possible by its flexible architecture. The talk focuses mostly on example usages and less about their implementation. Specifically we will discuss and showcase the following: - lazy evaluation with the thunk object space - lightweight threadlets similar to those provided by stackless python - Oz-like dataflow variables with the logic object space - Writing compatible extension modules for PyPy and CPython with the PyPy extension compiler - The Javascript backend and its uses</p>

## **Python in Science - 40-SS-D01 (04 July 11:00-12:40)**

time [id] title

11:00	<p><b>[77] LLG Python: A Python based approach for high performance Parallel Geometric Computing (00h30')</b></p> <p><i>Speaker: Mr. FARRET, Jérémie</i></p> <p>LLG Python defines a Python based programming paradigm for a new generation of geometric simulation, able to unify volumic data representation and processing for real time applications. It unifies in particular geometric and graphic pipelines, replacing parametric and polygonal surfaces with descriptive polynomials for generalized algebraic surfaces. This approach virtually enables scientific simulation interoperability for domains as various as robotics, GIS and territorial simulation, aeronautics or naval simulation. LLG Python's high performance, Parallel Geometric Computing environment enables universal 3D programming to bring an answer current simulation challenges and limitations.</p>
11:35	<p><b>[85] The Snake In The MeqTree: Using Python For Simulation And Calibration Of Radioastronomical Observations (00h30')</b></p> <p><i>Speaker: Dr. SMIRNOV, Oleg</i></p> <p>Calibration of observational data from the next generation of radio telescopes presents a number of unique challenges, not only in terms of pure computational volume, but also in terms of managing complexity. In general, the calibration process involves constructing a combined model of the instrument and the observed sky, and fitting that model to the observations. Due to the great variety of sky sources and instruments, specific models can be very elaborate, but also very different from each other. The MeqTree package attempts to address this problem by providing a very flexible model construction &amp; fitting toolkit which, in principle, should be applicable to all existing and future instruments. Using MeqTrees, inscrutably complicated and detailed models ("trees") may be constructed. Python plays two critical roles in taming that complexity: * TDL (Tree Definition Language), which is the language used to build up our models, is underpinned by Python. TDL allows for very rapid development of trees for simulation and calibration. * The MeqBrowser, written completely in Python/PyQt, is a graphical browser for running, examining and debugging the trees. The talk will provide a brief overview of radio interferometry, describe the calibration and simulation problems we face, and show how Python has enabled some crucial breakthroughs in the process.</p>
12:10	<p><b>[118] Python binding for Geant4 toolkit using Reflex/PyROOT (00h30')</b></p> <p><i>Speaker: Dr. POKORSKI, Witold</i></p> <p>Geant4 is one of the principal toolkits used for the simulation of the passage of particles through matter. It is implemented in C++ and it exposes to the users a set of methods (of different classes) needed to construct a functional application. In other words, users need to interact with Geant4 objects via a C++ API and therefore users' Geant4 based applications are normally written in C++. In this paper we discuss how to use the Reflex/PyROOT tools to create a Python binding for the Geant4 classes. Such a binding gives the users the possibility of implementing their applications in Python as well as to interact and configure them from the Python prompt. Moreover, using the already existing Python binding for the ROOT data analysis framework, the Python binding for Geant4 allows to run interactively and simultaneously the two applications from the Python prompt. This proves to be very interesting, especially in the case of development and debugging of simulation applications. The advantages of using the Reflex/PyROOT tools for creating a Python binding for C++ classes will be presented in this paper.</p>



**Web Frameworks - 40-S2-B01 (04 July 11:00-12:40)**

time [id] title

11:00 [117] Frameworks Shootout (01h40')

**Business and Applications - 40-SS-C01 (04 July 14:00-15:40)**

time [id] title

14:00 [121] "What not how" - Questionnaire development with Zope &amp; Python (00h30')

*Speaker: Mr. CLARK, Charlie*

Computers are well-suited for routine, recurring tasks such as data entry. In fields such as medicine effective use can lead to considerable improvements in productivity and patient care if doctors spend less time asking routine questions. As part of a research project at the University of Aberdeen I have been investigating possible systems. Any system should require as little maintenance by programmers as possible and should be configurable by managers, although the data captured would only be required for a doctor-patient interview, it should be stored in a reusable form. While there are several very questionnaire systems around they are generally geared towards statistical evaluation. Furthermore, while providing a great deal of flexibility in questionnaire and report layout, the data essentially remains locked within the application. It was decided to develop a prototype to support a generic, data-driven approach with a RDBMS backend for connection with other systems. To provide the greatest degree of flexibility the core part of the application requires very little programming. Zope, Python and PostgreSQL provide a good environment for this kind of requirement: PostgreSQL ensures data integrity, Python is well-suited for writing business rules, Zope is an excellent environment for integration. As a result most problems are of a conceptional nature. Due to the highly specific nature of parts of the project, it is unlikely that the software will be released.

14:35 [87] On the usage of Python in the CERN Document Server's digital library and conference management tools (00h30')

*Speakers: Mr. BARON, Thomas, Mr. LE MEUR, Jean-Yves*

Python has been intensively used at CERN since 2001 by the CERN Document Server team to provide services in two areas: - Digital document management within the CDS Invenio application, running on top of MySQL DB - Digital conference management within the CDS Indico application, running on top of a Zope DB; and currently used to run EuroPython 06! In this presentation, after providing an overview of the usage and architecture of these two open source packages, we will describe the technical options that have been selected after comparison with other solutions. The choice of Python/MySQL versus Python/ZODB will be discussed. Finally, the role of the GNU Public Licensing in the evolution of CDS software with the creation of a world-wide consortium of developers will be underlined.

15:10 [95] TinyERP: opensource enterprise management software (00h30')

*Speaker: Mr. DE MENTEN, Gaëtan*

Tiny ERP is a complete ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) package. Its main features include accounting (analytic and financial), production management (MRP), stock management, sales and purchases management, and much more. Technical features include flexible workflows, an object relational mapping (on top of PostgreSQL), a dynamically built GUI, an XML-RPC interface, and reports designable using OpenOffice (final output is in PDF). This talk will give a brief introduction about what Tiny ERP is, what it can do and how it works. The presentation will be divided in two parts. The first part will present the functionalities and end with a short live demo. The second part will focus on the technical aspects. It will include a presentation of the framework, how the different technologies involved (Python, PostgreSQL, XML-RPC, GTK, ...) are used, and some live demonstration of simple customization.

**Python Language and Libraries - 40-S2-A01 (04 July 14:00-15:40)**

time [id] title

14:00 [61] Using decorators (00h30')

*Speaker: Dr. SIMIONATO, Micheles*

I will talk about: - what decorators are (with examples); - how to write custom decorators; - when to use them and when not to use them; - especially cool decorators; - 5 minutes of discussion about the decorator module I wrote.

14:35	<b>[101] Useful and New Modules (00h30')</b> <i>Speaker: DALKE, Andrew Dalke</i> Python comes with batteries, and more are added with every release. It's hard to keep track of all of them, and I won't even try. My talk will cover some of the standard library modules I think more people should know about. I'll cover the new ctypes and ElementTree modules and older modules including subprocess, csv, and optparse. Depending on the time I'll also show examples using the bisect, heapq, and textwrap modules.
15:10	<b>[91] py.execnet: ad-hoc networking (00h30')</b> <i>Speaker: KREKEL, holger</i> Jan Balster presents a distributed computing approach and applications. There are different approaches to the distribution problem, each solution emphasises a special part of the problem. py.execnet provides an ad-hoc approach towards distributing programs which can work without any server side installation (except SSH and Python installed). It executes client-provided code fragments asynchronously, making server code maintenance superfluous and enables the clients to define the communication protocols. As part of the py lib py.execnet is developed in a test-driven style and is used in many different projects. After a quick introduction to using py.execnet this talk will provide interesting examples like robust hotsyncing of svn repositories or collecting system information from many (XEN) hosts.

## Web Frameworks - 40-S2-B01 (04 July 14:00-15:40)

time [id] title

14:00	<b>[24] A Lucene-based cataloging solution for Zope-3 and CPS-3.4 (00h30')</b> <i>Speaker: Mr. ANGUENOT, Julien</i> Lucene is an open source project from the Apache Software Foundation written in Java. This is a high-performance, full-featured text search engine library. NXLucene, from Nuxeo SAS, is a multi-threaded remote server handling Lucene stores. It takes advantage of the freaking fast PyLucene Python bindings and uses Twisted for its implementation. It uses some parts of the Zope3 component architecture as well. NXLucene currently supports the XML-RPC and SOAP protocols. (Its roadmap even includes an ICE connector for the 1.x branch.) As well, NXLucene might be seen as a good example of what could be achieved using the best parts of different worlds (Java Lucene, PyLucene, Zope3, Twisted,...). nuxeo.lucene is a cataloging component integrated to the Zope3 application server currently offering an xml-rpc proxy to an NXLucene remote server. As well, It offers an abstraction for Python objects' cataloging strategy providing the ability to specify how Python objects should be indexed and retrieved from a Lucene store through NXLucene. (It is important to note here that any remote server providing an XML-RPC remote interface could be theoretically used.) CPSLuceneCatalog is a CPS-3.4.x specific product adding the CPS specific business rules to nuxeo.lucene. For example, it takes care of the way different versions of CPS documents should be indexed. CPSLuceneCatalog is a complete substitute for the ZCatalog which is showing its limits when dealing with millions of objects within the ZODB. CPSLuceneCatalog will be shipped along with the next major release of CPS, version 4, along with the JackRabbit JCR repository. The result is a big win on large scale deployments: - Indexing and searching are much faster and more scalable compared to ZCatalog - Indexing and searching are much more powerful than with ZCatalog - (Analysis, ranking, etc...) - Zope global performance is increased because Zope no longer deals with the indexing and searching business. NXLucene is an open source project under the LGPL. nuxeo.lucene is an open source project available under the ZPL. CPSLuceneCatalog is an open source project available under the GPL. For more information: - CPS Project: <a href="http://www.cps-project.org">http://www.cps-project.org</a> - Zope3: <a href="http://dev.zope.org/Zope3">http://dev.zope.org/Zope3</a> - Apache Lucene: <a href="http://lucene.apache.org/java/docs/index.html">http://lucene.apache.org/java/docs/index.html</a> - PyLucene: <a href="http://pylucene.osafoundation.org">http://pylucene.osafoundation.org</a> - NXLucene: <a href="http://www.cps-project.org/sections/projects/nxlucene">http://www.cps-project.org/sections/projects/nxlucene</a> - nuxeo.lucene: <a href="http://svn.nuxeo.org/trac/pub/browser/Zope3/nuxeo.lucene/trunk">http://svn.nuxeo.org/trac/pub/browser/Zope3/nuxeo.lucene/trunk</a> - CPSLuceneCatalog: <a href="http://svn.nuxeo.org/trac/pub/browser/CPS3/products/CPSLuceneCatalog/trunk">http://svn.nuxeo.org/trac/pub/browser/CPS3/products/CPSLuceneCatalog/trunk</a> - Nuxeo SAS: <a href="http://www.nuxeo.com">http://www.nuxeo.com</a>
14:35	<b>[63] CPSSkins: theme editor for Zope3 (00h30')</b> <i>Speaker: Mr. ORLIAGUET, Jean-Marc</i> CPSSkins is an application that allows site designers to compose pages by selecting and arranging UI elements on a canvas without having to use a programming language. This talk goes through the new exciting features found in the upcoming zope3 version of the application such as: - the new AJAX-based user interface - the WYSIWYG style editor - a simple API for creating custom portlets and widgets - the use of "perspectives" to control the visibility of elements on a page - the ability to create sites through-the-web and on the file-system - the XML import/export facility - the highly modular rendering engine - compatibility with Zope2/Five - ... The talk is targeted towards both developers and site designers. More information about CPSSkins can be found at <a href="http://www.z3lab.org">http://www.z3lab.org</a>
15:10	<b>[96] Panel: Getting Started with the Zope Foundation (00h30')</b> <i>Speaker: Mr. FULTON, James</i> The Zope Foundation was legally formed on March 25, 2006. The first Foundation board will be elected in June 2006. This panel, consisting of Foundation Board members or their representatives, will present progress on forming the Foundation and, more importantly, solicit input and volunteers from members of the Zope community present.

**Lightning Talks - Main Auditorium (04 July 16:00-18:00)**

time [id] title

16:00	[119] <b>Lightning Talks (02h00')</b>
-------	---------------------------------------

**Keynote - Main Auditorium (04 July 18:30-19:30)**

time [id] title

18:30	<b>[31] The Future of Python (01h00')</b>
-------	---

*Speaker: Mr. VAN ROSSUM, Guido*

The next major version of Python, nicknamed Python 3000 (or more prosaically Python 3.0), has been anticipated for a long time. For years I have been collecting and exploring ideas that were too radical for Python 2.x, and it's time to stop dreaming and start coding. In this keynote I will present the community process that will be used to complete the specification for Python 3000, as well as some of the major changes to the language and the remaining challenges.

## Wednesday 05 July 2006

### **Business and Applications - 40-SS-C01 (05 July 09:00-10:05)**

### **Refereed Papers - 40-SS-D01 (05 July 09:00-10:40)**

time [id] title

09:00	<b>[19] Pyphant - A Python framework for modelling reusable data processing tasks (00h30')</b> <i>Speaker: Mr. ZIMMERMANN, Klaus</i> We present the Open Source framework Pyphant for the creation and application of data flow models. The central idea of this approach is to encapsulate each data processing step in one unit which we call a worker. A worker receives input via sockets and provides the results of its data processing via plugs. These can be inserted into other worker's sockets. The resulting directed graph is called a recipe. Classes for these objects comprise the Pyphant core. To implement actual processing steps Pyphant relies on third party plug-ins which extend the basic worker class and can be distributed in so-called Pyphant-worker-archives (PWA). On top of the core, Pyphant offers a data exchange layer based on scipy arrays and PIL images which facilitates the interoperability of the workers. A third layer comprises textual and graphical user interfaces. The latter allows for the interactive construction of recipes, the former for the batch processing of data. Our contribution discusses the Pyphant framework and presents example recipes for certain scientific data analysis tasks taken from image processing and the solution of ill-posed problems.
09:35	<b>[23] Deep Integration of Python with Semantic Web Technologies (00h30')</b> <i>Speaker: Mr. BABIK, Marian</i> The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate the information available on the Web. The Semantic Web will build on the well known language stack, part of which is the Web Ontology Language (OWL). Semantic Python (SETH) is a software effort to deeply integrate python and the description logic (DL) subset of the OWL, i.e. extend python to seamlessly support a logic programming paradigm. The deep integration of both languages introduces the notion of importing the OWL-DL into the programming context so that OWL classes can be used alongside classed defined normally. In this article we present a metaclass-based implementation of the deep integration ideas, which is a promising way of achieving such integration. The implementation is an early Python prototype supporting in-line class and property declaration, instance creation and simple triple-based queries. The implementation is backed up by the well known OWL-DL reasoner Pellet. The integration of the Python and OWL-DL through metaclass programming provides a unique approach, which can extend the current python-based web frameworks and provide the needed support for the Semantic Web technologies.
10:10	<b>[21] Implementation of a Transaction-based Peer-to-Peer Protocol (00h30')</b> <i>Speaker: Ms. NYGÅRD, Sofia</i> The paper describes a lightweight implementation of a peer-to-peer protocol. The implementation is intended as a testbed for ideas concerning the utilization of the peer-to-peer paradigm. The protocol takes the Gnutella protocol as a starting point, although the present implementation differs from the Gnutella protocol regarding intent as well as routing solutions. The protocol is implemented in Python since various data types and modules offered by Python make it possible to limit the code to a manageable size. The GUI is implemented with Tkinter. The design has two substantial features. Firstly, the protocol is transaction- based. As a consequence, the pong messages are no longer required since every successfully transferred message, independent of type, is a sign that a node is active. An adaptive ping algorithm further reduces the load on bandwidth. Secondly, there is only one entry point to the router. The way in which the router communicates with the GUI is symmetrical to the way in which it communicates with the network. This means that the connections between the router and the GUI are characterized by socket communication and protocol messages. This results in a fairly independent user interface, which creates a basis for further utilization of peer-to-peer systems.

### **Teaching - 40-S2-A01 (05 July 09:00-10:40)**

time [id] title

09:00	<b>[76] Python Academy - Teaching Python in Germany (00h30')</b> <i>Speaker: Dr. MÜLLER, Mike</i> Python Academy is a teaching firm that specializes in Python and Python related topics. It offers courses for beginners, programmers in other languages and scientists. It also provides Python-Trainers the opportunity to teach at its training center in Leipzig, Germany. Teaching languages are German and English. Besides offering courses it hosts the Leipzig Python User Group at its monthly meetings. It is the leading organizer of a Python-Workshop in September of this year in Leipzig. Other activities include speeches at different occasion about Python and contributions to a translation of an English open source book into German. Python Academy participated in initiative "Informatikjahr" ( <a href="http://www.informatikjahr.de/">http://www.informatikjahr.de/</a> ) by the German Ministry for Education and Research. with the slogan "Specialists as Programmers". Experiences in combining teaching in the business sector and community activities will be presented. Python Academy is seeking collaboration with Python trainers that are interested in teaching their special Python topic in Germany.
09:35	<b>[15] The African Python: Programming and Training in South Africa (00h30')</b> <i>Speaker: Prof. JOUBERT, Fourie</i> Python has become the language of choice for bioinformatics, especially in teaching bioinformatics students to program. Bioinformaticists students come from various backgrounds including biology, computer science, mathematics, statistics and physics. While many of these disciplines encourage programming skills and structured thinking, biologists often have a hard time adapting to the world of computer programming when immersed at post-graduate level. Python programming and training in South Africa will be discussed broadly, followed by specific reference to the strategies followed by the South African National Bioinformatics Network community in their integrated national training programme. The landscape of Python software development for Bioinformatics in South Africa will also be introduced.
10:10	<b>[123] Liberlab, a pocket lab for all (00h30')</b> <i>Speaker: Mr. SCHNELL, Francois</i> The Open Source project Liberlab aims to promote scientific experimentation through the creation and use of a digital lab at a very low cost ( 15 €/€\$). Its modular and open design encourages original uses, particularly in the field of education, and can also be used to learn about robotics, automation, human- machine interface or interactive arts. The heart of the project is based on the ongoing improvements of microcontrollers. These low price integrated circuits are like tiny and autonomous "computers" used in many devices (DVD players, cars, etc.). Instead of using them in an autonomous way Liberlab combines the computing power of a PC with the commands and measuring capabilities of the microcontroller. Liberlab is available under the free GPL licence with a GUI application (Tkinter) and a Python module to allow users to build their own programs using Python. The presentation will explain the technological choices for this project and show examples of usages with Python modules like Vpython, Xtturtles and CherryPy.

## **Web Frameworks - 40-S2-B01 (05 July 09:00-10:40)**

time [id] title

09:00	<b>[3] Templating Systems Compared and Contrasted (01h05')</b> <i>Speaker: Mr. WITHERS, Chris</i> This talk is going to compare and contrast the various approaches available for templating in python web frameworks. A brief introduction to some of the main flavours will be given along with a discussion of their relative strengths and weaknesses. With luck, the audience will leave with a clearer idea of the options available to them!
10:10	<b>[45] JSON-RPC - makes web services communication simple (00h30')</b> <i>Speaker: Mr. KOLLHOF, Jan-Klaas</i> For web/browser-based applications/AJAX-applications it is important to efficiently communicate to web services to exchange data. There are existing protocalls, e.g. SOAP the not so Simple Object Access Protocol, that are rich in features but rather complex. This complexity is not needed for a lot of use cases. This is where JSON-RPC fits into the picture. It is a simple yet powerful RPC protocol which defines how services and clients should communicate with each other.

## **Agile Development - 40-SS-C01 (05 July 11:00-12:40)**

time [id] title

11:00	<p><b>[86] Kill -1: process refactoring in the PyPy project (00h30')</b></p> <p><i>Speaker: Mrs. DURING, Beatric</i></p> <p>In this talk we will present the experiences of running a collaborative OSS project within a EU-funded consortium structure. We will briefly present the initial decision process for integrating the community and the commercial aspects of the PyPy project. We will show how agile practices within the development team, primarily sprint-driven methodology resulted in a need for process refactoring and how a new agile management approach was implemented. We will spice up the analysis of the main drivers behind this change by referring to Hershey-Blanchards Situational Leadership model.</p>
11:35	<p><b>[84] Users' participation in the design process of Open Source Software communities: the case of Python (00h30')</b></p> <p><i>Speaker: Ms. BARCELLINI, Flore</i></p> <p>In this talk, we aim at analyzing the ways members of open-source software communities, especially users, participate in the software design process. The general objective of this research in cognitive ergonomics is to understand activities occurring during the design process in order to better support collaborative work, greater interaction between various stakeholders, and construction of design projects memory. In our current work, we focus on the Python community. In a first study, we analyze online discussions between participants engaged in the Python Enhancement Proposal or PEP design process. We propose an alternative approach to threading to represent online discussions: it is based on the quoting practices as a relevant link between messages. We also show the influence of participants status in design-oriented online discussions and highlight the evaluative nature of design activities, i.e. pair reviewing mechanisms. A second study, still in progress, aims at understanding the "Python galaxy", the links between the different communities in this galaxy, and the language evolution process from the idea of language evolution in one particular application domain, through its acceptance by one or several communities, to the PEP and its related modules implementations. The analysis of a "pushed-by-users" PEP extended design process shows how users may participate in the language evolution process, in particular through mechanisms of cross-participation between communities. This talk will give us the opportunity to get feedback from the Python community on our research and could engage interested Python members into a reflective approach about their collaborative practices.</p>
12:10	<p><b>[14] From agile development to agile evolution of enterprise systems (00h30')</b></p> <p><i>Speaker: Dr. SAMARIN, Alexander</i></p> <p>The aim of this talk is to share my experience in the use of a practical architectural framework for the improvement of enterprise business systems. This framework uses the dynamic language Jython and agile development practices. This presentation is complementary to my presentation at the Plone conference 2005 in Vienna ("The use of Plone for enterprise solutions") – it covers the technical and practical aspects of agile evolution of enterprise systems.</p>

### **Misfits - 40-SS-D01 (05 July 11:00-12:40)**

time [id] title

11:00	<p><b>[92] subversioned system configuration (00h30')</b></p> <p><i>Speaker: Mr. KREKEL, holger</i></p> <p>Holger Krekel presents "vadm", an open-source tool enabling full non-intrusive versioning of unix system configuration files and directories. Multiple administrators can collaborate and communicate over the setup of one machine. Under the hood, vadm uses subversion and the "py lib" and is easy to learn if you know how to use the "svn" command line already. The talk will also present future directions like supporting remote management of a cluster of machines.</p>
11:35	<p><b>[50] Come on and plug me in! Writing and testing plugins for Bazaar (01h05')</b></p> <p><i>Speakers: Mr. COLLINS, Robert, Mr. ALLOUCHE, David</i></p> <p>Bazaar is a revision control system written entirely in Python. Programmers can use Bazaar's plugin system to provide new options, features and facilities. We demonstrate writing a plugin for Bazaar and describe what having a plugin system has done to and for Bazaar.</p>

### **Teaching - 40-S2-A01 (05 July 11:00-12:40)**

time [id] title



11:00	<b>[17] Vision: Visual Software IntegratiON (00h30')</b> <i>Speaker: Prof. SANNER, Michel</i> <p>Understanding the amazingly complex biological systems sustaining life requires integrating, analyzing and understanding huge amounts of data. Computers are the tools of choice for such a task. However, profound knowledge of biology is needed for analyzing the data and learning from it. It is impossible to foresee the multitude of ways data can be visualized or computational methods can be combined to test hypotheses and often, it is assumed that in order to use computers to their full potential, scientists have to become programmers. We have explored the visual programming paradigm as a way to empower scientists with the ability to interactively program applications without having to worry about data structures or syntactical details of a programming language (even as simple and clean as Python). We have developed Vision: a software component that supports visual programming, and written adapters exposing the functionality of many Python packages in this environment. The concept of visual programming is not new however, the interpretive nature of the Python language allows for powerful and elegant solutions to many limitations witnessed in other similar tools. In this paper we will present Vision, highlight fundamental differences with other tools, and discuss benefits and challenges associated with this approach.</p>
12:10	<b>[100] xturtle - an extended turtle module for Python as a vehicle for teaching programming concepts (00h30')</b> <i>Speaker: Mr. LINGL, Gregor</i> <p>A new extended Tkinter based turtle module will be presented: (1) Motives and pedagogical goals. (2) The design of the extended module. (3) The underlying architecture especially in respect of possible extensions and portings. (4) A set of sample scripts covering a wide range from elementary to rather sophisticated. (5) turtle graphics as a backbone for an introductory programming course/book. Finally (6) a discussion will be initiated about the presented module as a starting point for a replacement for the existing turtle module of the Python standard distribution.</p>

## **Web Frameworks - 40-S2-B01 (05 July 11:00-12:40)**

time [id] title

11:00	<b>[103] Zope: The June releases (00h30')</b> <i>Speaker: Mr. VON WEITERSHAUSEN, Philipp</i> <p>It's the second time that the Zope project has made releases according to the 6 month release schedule. Zope 2.10 and Zope 3.3 come with numerous new features, improvements and refactorings. This talk will give an overview of the changes in the "June releases" and point out some planned features for the upcoming release cycle.</p>
11:35	<b>[9] Tramline, big files are fun! (00h30')</b> <i>Speaker: Mr. VLOOTHUIS, Jeroen</i> <p>We at Pareto have created a streaming video Portal on top of Plone. It is created utilizing Plone's CMS capabilities combined with other open source packages. Because large binary data (like videos) can hurt the performance of the Zope database (ZODB) we set up Tramline. Tramline is a mod_python program which filters incoming post data. When it notices a file upload it extracts the data from it and puts that into a file. Then it replaces the post body with an id matching the filename. Upon a get it filters the response headers to see if the body is a Tramline id. In case it is it will replace the response body with the contents of the file stored on disk. This way Tramline ensures all binary file data is kept on disk instead of in the database. Yet it makes development and testing without it no different than you would normally. In my talk I would like to explain the principles and usage of Tramline. I will do this using our use case as a backdrop.</p>
12:10	<b>[67] Linktally: Ranking popular pages in a CMS (00h30')</b> <i>Speaker: Mr. PIETERS, Martijn</i> <p>Linktally works around typical CMS roadblocks like high-read, low-write databases, clustering and extensive caching to provide a approximation to page popularity.</p>

## **Agile Development - 40-SS-C01 (05 July 14:00-15:40)**

time [id] title

14:00	<b>[69] Managing the Launchpad team at Canonical (00h30')</b> <i>Speaker: ALEXANDER, Steve</i> <p>Steve Alexander talks about managing the Launchpad team: 16 Python programmers from around the world working on a complex database application. Topics include: Effective IRC meetings, Conference VOIP calls, Code review, Distributed source code management, Automated testing, Sprints and meetings.</p>
-------	--

14:35	<b>[102] Managing a distributed company (00h30')</b> <i>Speaker: WAGNER, Lene</i> This talk will discuss the experiences with implementing agile and distributed work methods in a small company that is involved in various international projects. The vision of working collaboratively, distributed and with shared responsibilities somehow contradicts contractual commitments and customer relations. Lene Wagner, manager of the distributed open source company merlinux, describes experiences and methods how to combine this within the commercial context of a small enterprise. The talk discusses shared decision models and peer-review strategies and how to manage outer and inner commitments. It deals with how to implement formal structures serving both agile processes and legal requirements. The talk will also describe problems, methods and findings of handling a reliable distributed communication via virtual channels.
15:10	<b>[114] Agile customer management: successes and failures (00h30')</b> <i>Speaker: Ms. KESMINAITE, Aiste</i> I will talk about our company's experience in applying certain XP practices when dealing with clients, how different clients understand these practises and react to them and what kind of problems we have trying to explain and apply XP. About the author: Aistė Kesminaitė is the managing director of a young but already distinguished team of software engineers from Lithuania. She and her team use XP and other agile practises for most aspects of the business.

### **Misfits - 40-SS-D01 (05 July 14:00-15:40)**

time [id] title

14:00	<b>[41] SCREWS - An Architecture For Simulating Errors in Web Servers (00h30')</b> <i>Speaker: Mr. ZADKA, Moshe</i> SCREWS is a flexible architecture for simulating errors in web servers, intended for testing of error detection and recovery tools. The flexibility comes from the ability to write "SCREWSlets", tiny Python scripts which determine whether to send an error, and what kind.
14:35	<b>[88] Dual Scripting in a Virtual Reality Engine. Embedding Python in XVR (00h30')</b> <i>Speakers: Mr. RUFFALDI, Emanuele, Mr. APRILE, Walter</i> In this talk we present the embedding of Python inside the eXtreme Virtual Reality (XVR) Engine. XVR is an advanced and lightweight system for the development of Virtual Reality applications both for the Web and for 3D Immersive systems. This system has been extensively used in European Projects for the construction of high quality 3D graphic applications enhanced with VR devices. XVR applications are written in the S3D scripting language and compiled into a bytecode representation executed by the XVR Virtual Machine. The scripting language and the virtual machine are specialized and optimized for the manipulation of 3D entities and efficient object oriented programming. The XVR system allows to develop and deploy 3D application easily by asking the developer only to write the core rendering loop and place the application in a Web page. The XVR Engine provides natively a set of classes for creating Virtual Reality applications ranging from 3D graphics (both high level classes and low level OpenGL functions), 3D sound and physics. Moreover it is possible to extend such features with external native modules providing the support for new devices, simulation engines or rendering techniques. At runtime the external modules are loaded and their classes integrated in the system namespace. PYXVR is a XVR extension that allows to embed a Python script inside a XVR application or, putting it in another way, write a Virtual Reality application using the XVR Engine features. Typically the XVR application loads the PYXVR extension, executes the scripts and invokes Python in response to the rendering (OnFrame) and animation (OnTimer) events of the XVR Engine. The result is a dual script environment in which the two scripting engines cooperate for providing the best of the two worlds. The Python scripts are able to access all the functions and classes defined by XVR (both scripted and native), while the XVR script accesses the Python scripts through the eval and callback functions. There are some advantages to using Python inside a XVR application. First there is the possibility of modifying and extending the application using the script, because the XVR application is completely defined at compilation time. Then there is the possibility of using the Python libraries and multi-threading for enhancing the XVR application. Despite the advantages there are some drawbacks that should be taken into account. First there is the aspect of security. XVR applications are typically executed in the Web Browser and the XVR classes respect the security requirements of this environment. The PYXVR extension allows the unrestricted import of modules and for this reason it is currently limited to local XVR applications. The second disadvantage of PYXVR for 3D Web application is the weight of the Python engine whose core shared library is in the order of one megabyte, a size that is equivalent to the whole XVR engine comprising the Virtual Reality features. With regards to the execution speed, the S3D scripts are faster than the Python ones in managing objects and external entities because during the off-line compilation of the script all the names and functions are known and encoded in the byte-code. The simpler type space of S3D provides also additional optimizations when dealing with 3D entities. The talk discusses the overall PYXVR extension, the aspects of integration between the two environments, the multi-threaded features and finally a performance evaluation of this solution. Concluding, the PYXVR extension presented in this talk can be used by XVR developers for enhancing their XVR applications and at the same time by Python developers for the fast prototyping of Virtual Reality applications. Information about XVR can be found at <a href="http://www.vrmedia.it">http://www.vrmedia.it</a> More information on PYXVR will be shortly provided on: <a href="http://wiki.vrmedia.it/index.php?title=PYXVR">http://wiki.vrmedia.it/index.php?title=PYXVR</a>



**15:10 [106] Speed up your Python code (00h30')***Speaker: Dr. SCHWARZER, Stefan*

Does it seem your Python code (or even Python code in general) is too slow? If yes, come to this talk. It explains: - which rules should guide you when optimizing Python code - when you should optimize at all - how to find the bottlenecks in your code - what big-O notation is and what it's good for - which changes may help to speed up your Python code - using hardware, algorithms or Python-specific idioms. This talk is for developers of Python code. The less experience you have, the more you will profit. If you are familiar with software development but new to Python, especially the Python-specific tips will interest you.

**Web Frameworks - 40-S2-B01 (05 July 14:00-15:40)**

time [id] title

**14:00 [79] Generic Functions in Turbogears (00h30')***Speaker: Mr. BELAK, Simon*

Generic functions are a powerful form of abstraction promoting the DRY principle and in the case of frameworks lend themselves to declarative programming and allow users to seamlessly extend functionality with minimal boilerplate. In this talk common uses cases for generic functions in TurboGears will be presented based on concrete examples followed by a discussion on problems we encountered using generic functions, as implemented in RuleDispatch package from PEAK, both on framework and user level, and how these can be mitigated.

**Lightning Talks - Main Auditorium (05 July 16:00-18:00)**

time [id] title

**16:00 [120] Lightning Talks (02h00')**