BERLIN 2012
CONFERENCE
17th-19th October

Reshaping Calc for better performance

Kohei Yoshida, SUSE, Inc.

Introduction (Kohei Yoshida)

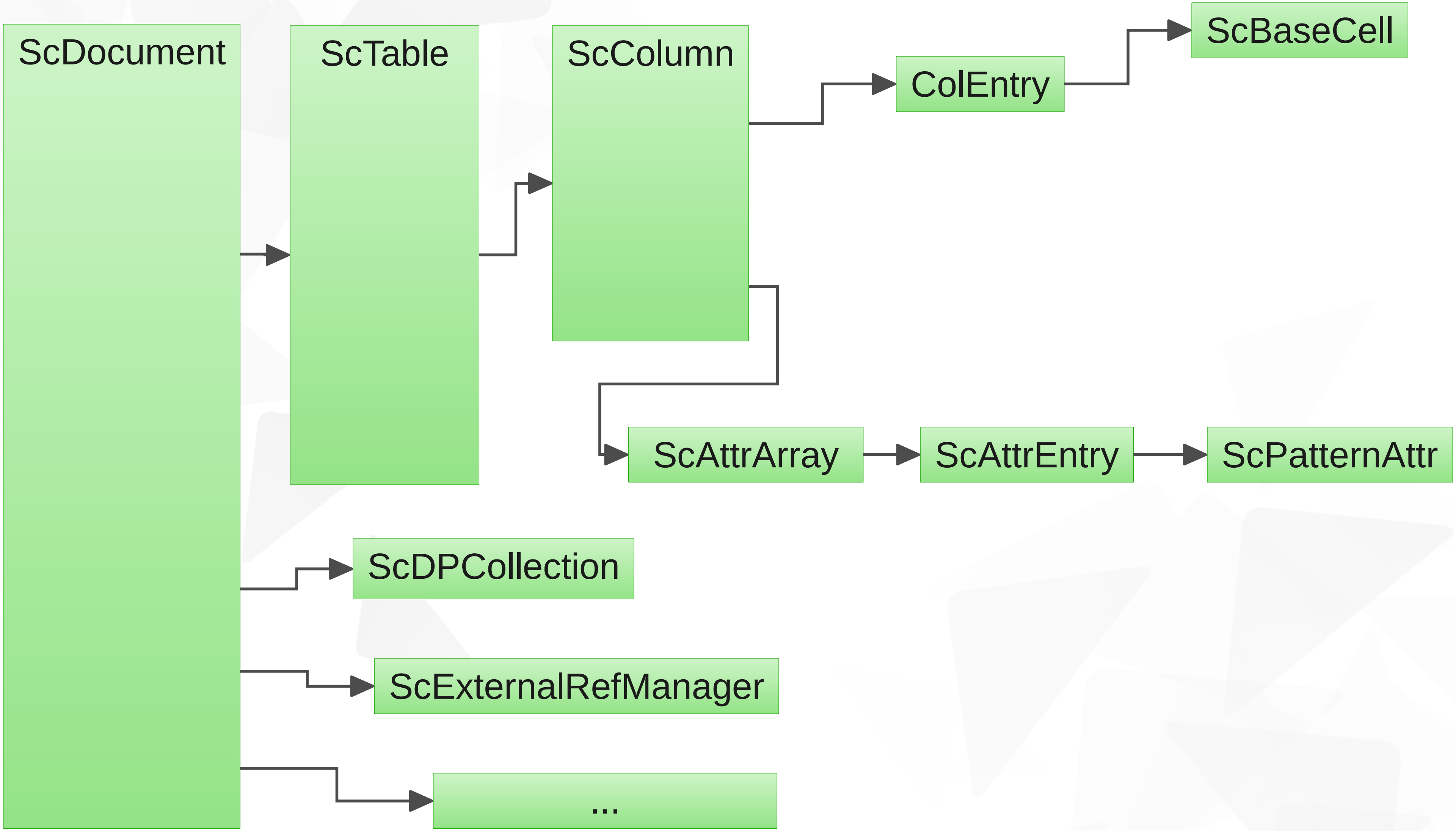
- Based in Raleigh, North Carolina. Originally from Japan.
- Spare-time hacker turned full-time.
- Hacking on OOo/LibO since 2004.
- Software Engineer at Novell since 2007 (later SUSE), with emphasis on LibreOffice Calc.
- Blog: http://kohei.us/
- Twitter: @kohei_yoshida
- G+: https://plus.google.com/u/0/107646708505179576030

Topics

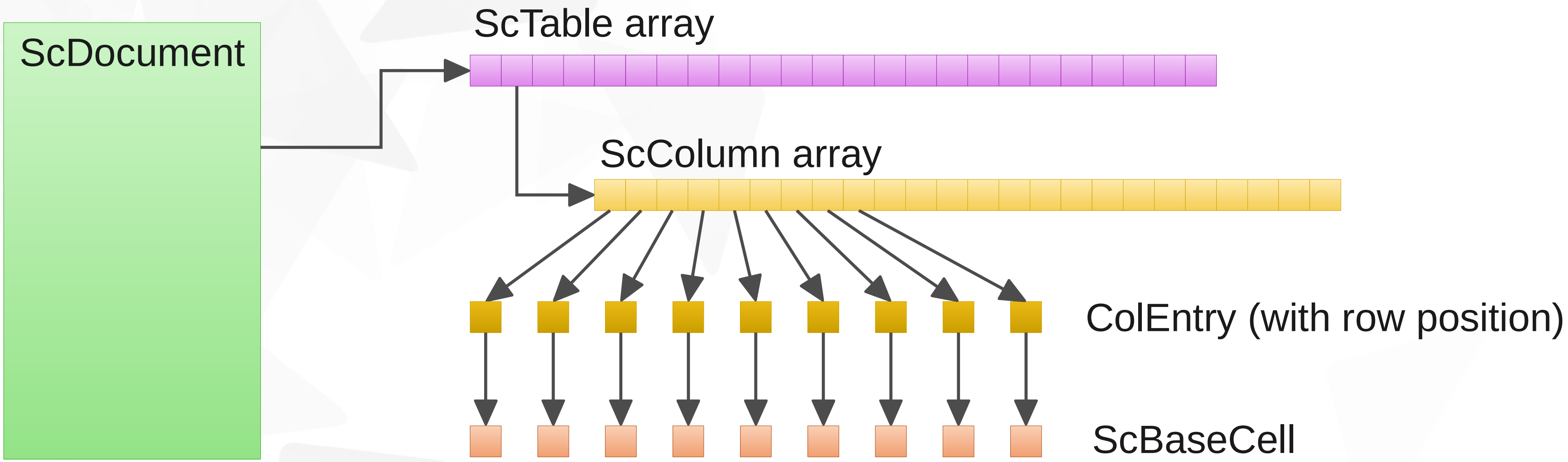
- **Document core**
  - Current structure
  - Ideal structure
  - Migration plan
- **Formula engine**
  - Current design & issues
  - Ixion - alternative engine
  - Short-term prospects
- **File Import filters**
- **Putting it all together**

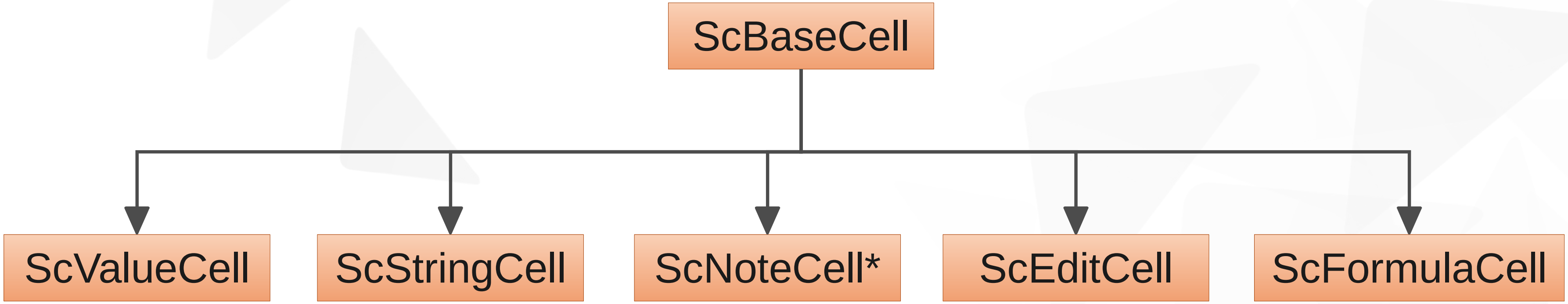
Document Core

Overview


ScDocument
ScTable
ScColumn
ColEntry
ScBaseCell
ScAttrArray
ScAttrEntry
ScPatternAttr
ScDPCollection
ScExternalRefManager
...

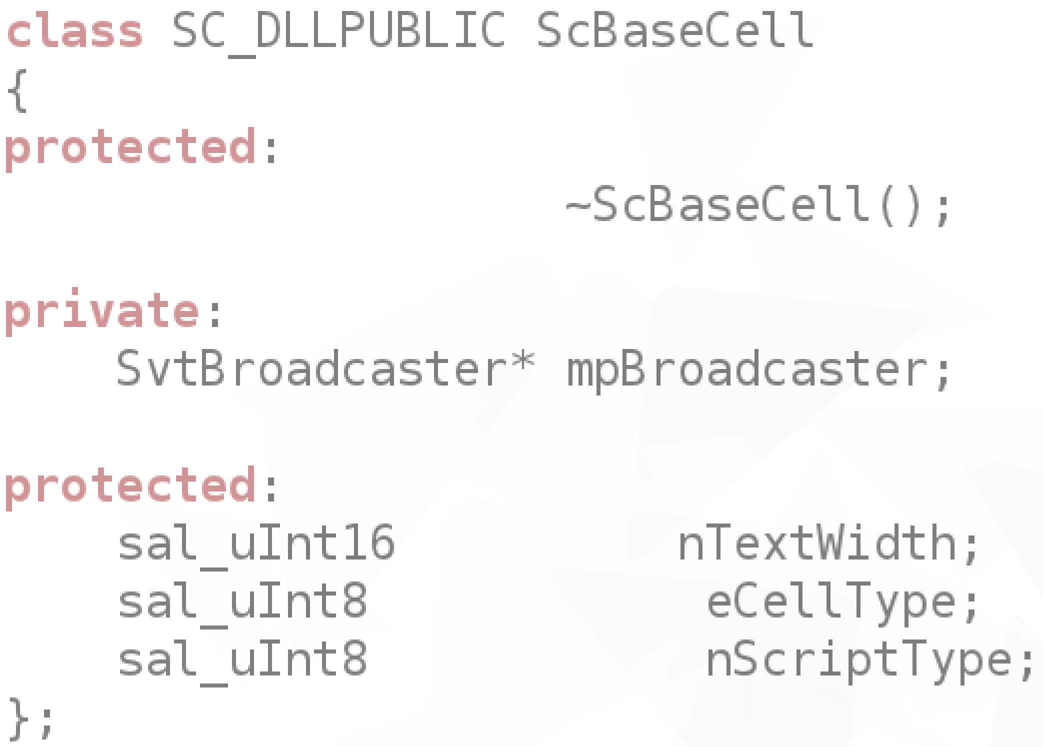
Cell storage





Good & Bad

- **The Good**
  - Intuitive – design very similar to how cells are organized on screen.
  - Polymorphic design true to the principle of object-oriented design.
- **The Bad**
  - Cell objects fragmented across multiple memory pages. Not ideal for formula engine's scalability.
  - Cell size bloat due to direct storage of peripherals.

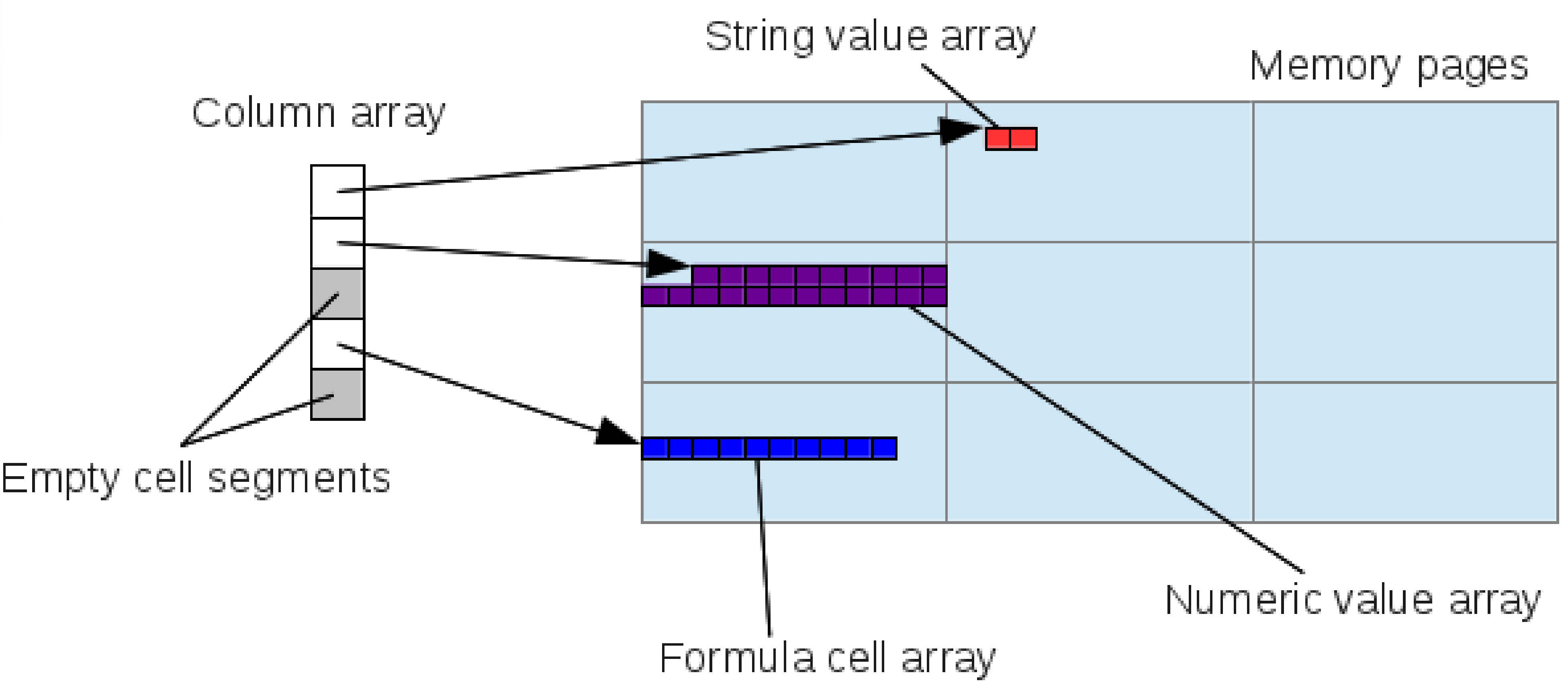


```
class SC_DLLPUBLIC ScBaseCell
{
protected:
                    ~ScBaseCell();

private:
    SvtBroadcaster* mpBroadcaster;

protected:
    sal_uInt16              nTextWidth;
    sal_uInt8                eCellType;
    sal_uInt8                nScriptType;
};
```

Alternative approach - array storage



http://kohei.us/2012/07/20/mdds-multi_type_vector-explained/

Benefit of array storage

- Space efficiency – each array stores only raw cell values.
- Better locality of reference – fewer memory pages to load when iterating through cell values.
- Further hardware acceleration – SIMD, GPU....
- Data structure already implemented in mdds as **multi_type_vector**. Usable for other storage needs.
  - Cell storage
  - Matrix storage
  - External reference cache

What we have done so far...

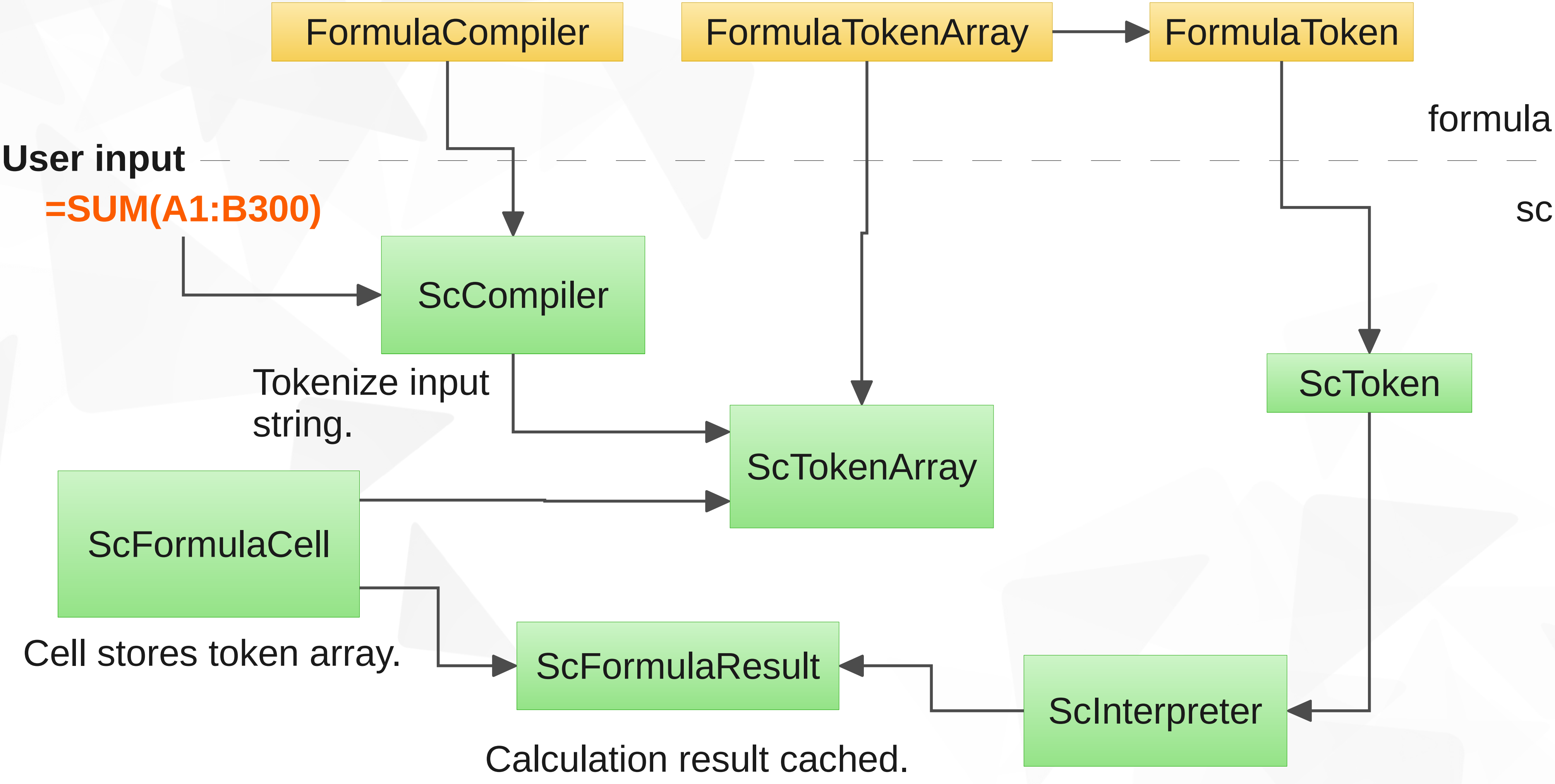
- Removal of ScBaseCell peripherals.
    - Notes – now in ScTable as ScNotes. (Markus Mohrhard)
- Base data structure implemented as mdds::multi_type_vector.

Array storage migeration plan

- Phase 1 – Numeric and string value storage
  - Move out **ScBaseCell** peripherals: broadcaster, text width, and script type → removal of **ScNoteCell**.
  - Remove use of **ScValueCell** and **ScStringCell** classes outside **ScDocument**.
  - Store raw numeric and string values in arrays.
- Phase 2 – Rich-text value storage (**ScEditCell**)
  - Remove use of **ScEditCell** outside **ScDocument**.
  - Store **EditTextObject** directly in arrays.
- Phase 3 – Formula cell storage (**ScFormulaCell**)
  - More complex. Come back later.

Formula Engine

Overview



=SUM(A1:B300)

Tokenize input string.

Cell stores token array.

Calculation result cached.

Good & Bad

- **The Good**
  - It works today.
  - Optimized for screen rendering.
- **The Bad**
  - Very complex beast.
  - The big split – sc and formula modules to cut the engine in half. Even more complex.
  - Listener & broadcaster pattern – no clear calculation order prior to calculation. Hard to parallelize.
  - Recursive calculation – stack memory bloat.
  - No unit tests.
  - Dependency on Calc core. Not re-usable.

Ixion (Alternative?)

- http://gitorious.org/ixion
- **Benefits**
  - Standalone C++ library, re-usable, unit test framework.
  - Multi-threaded interpreter.
  - Dependency relations resolved pre-calculation. Easier to parallelize.
  - Iterative calculation. No fear of running out of stack memory.
- **Issues**
  - Huge effort to match current engine.
  - Invasive change required.
  - Not everybody agrees with it.

Short-term strategy

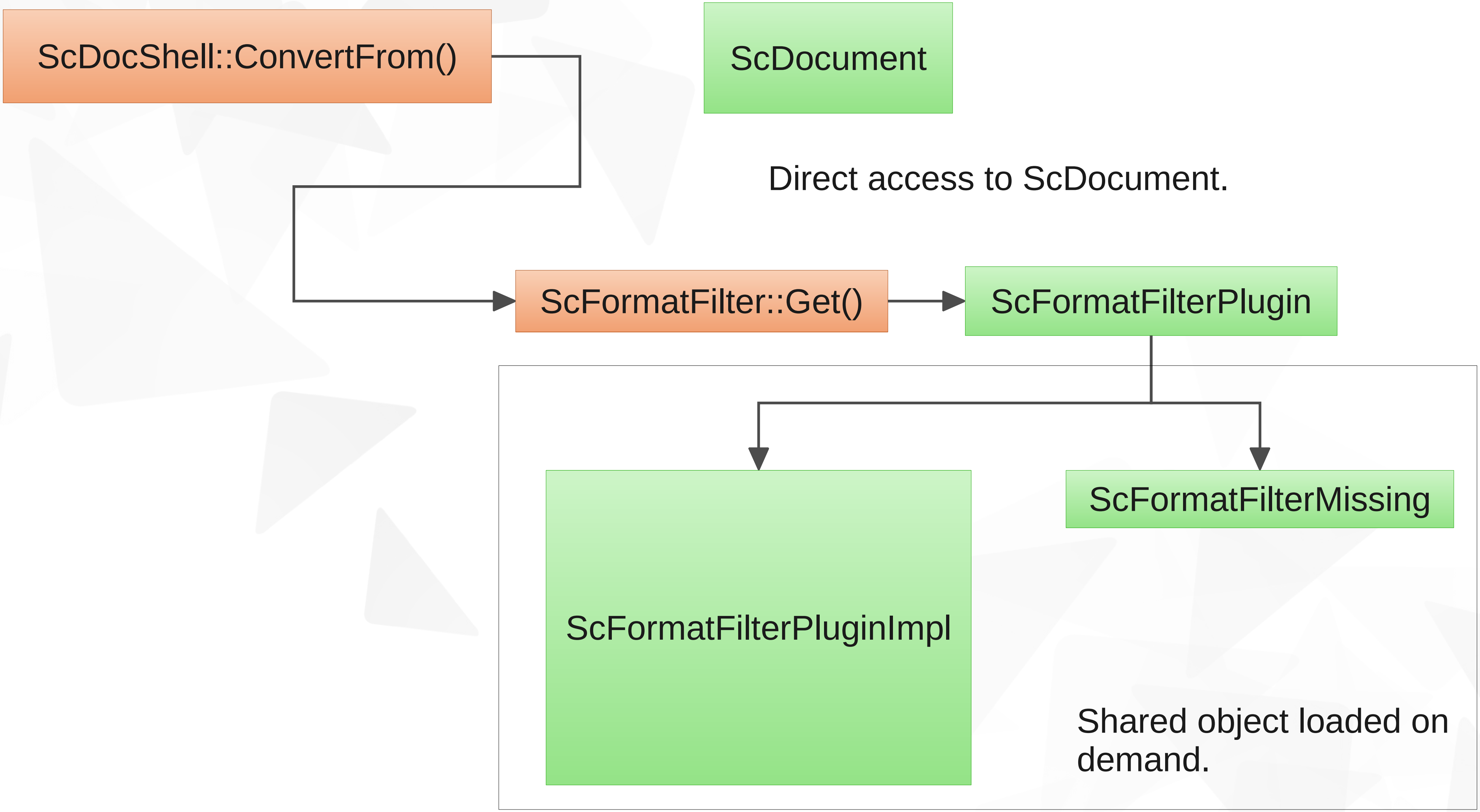
- Meet Ixion requirements that are beneficial on their own.
  - Application-wide shared strings.
  - Shared formula tokens.
  - Cells only store values.
- Keep improving the current formula engine.
  - Remove Calc A1 ODF specific special casing.
  - Range-based dependency tracking.
  - Database range reference syntax.
- Defer Ixion integration decision when Ixion is mature enough.

File Import Filters

Current situation

- **Calc internal import filters**
  - CSV, Lotus 123, Quattro Pro, Excel 4.0/5.0/95/97 (BIFF), dBase, DIF, SYLK, HTML, RTF
- **Pure UNO import filters**
  - XLSX (up to 3.5)
- **ODS import filter**
  - UNO XML parser
  - Access to ScDocument directly or via UNO API.
- **XLSX import filter (3.6 and later)**
  - UNO filter + access to ScDocument directly or via UNO API.
- **XSLT filters**
  - Not scalable at all. Poor performance.

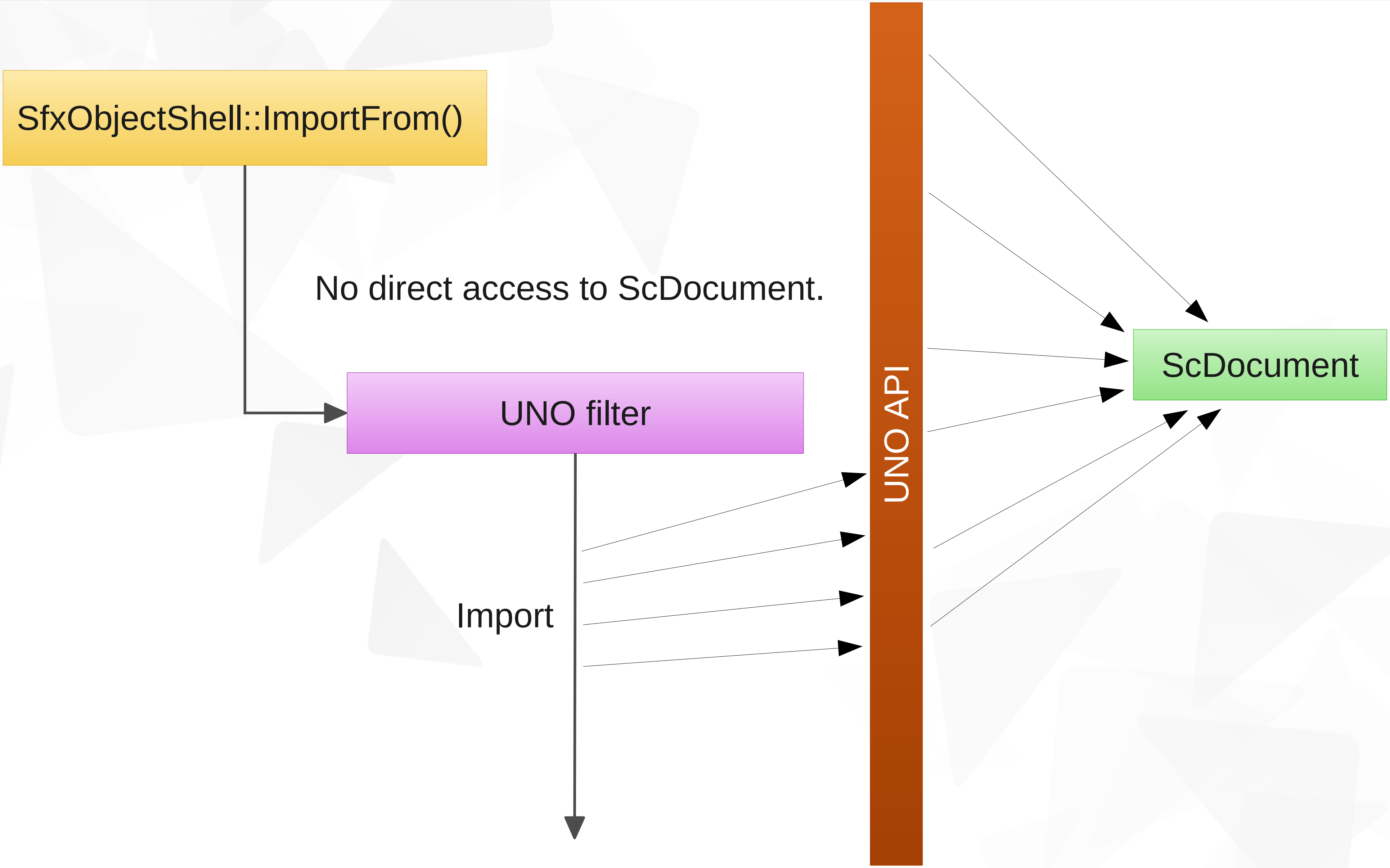
Overview (Calc internal import filters)



Direct access to ScDocument.

Shared object loaded on demand.

Overview (Pure UNO import filters)



SfxObjectShell::ImportFrom()

No direct access to ScDocument.

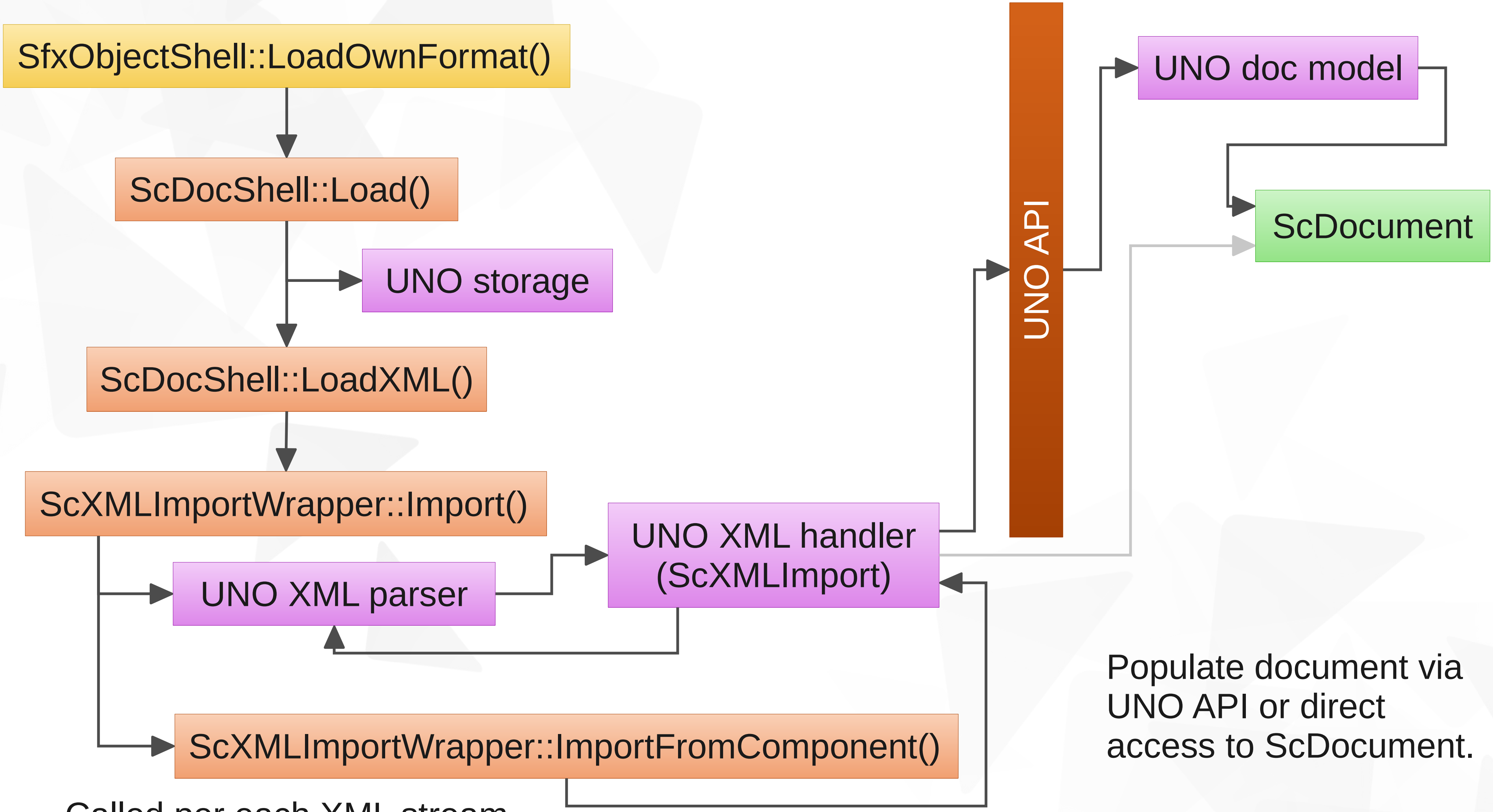
UNO filter

Import

UNO API

ScDocument

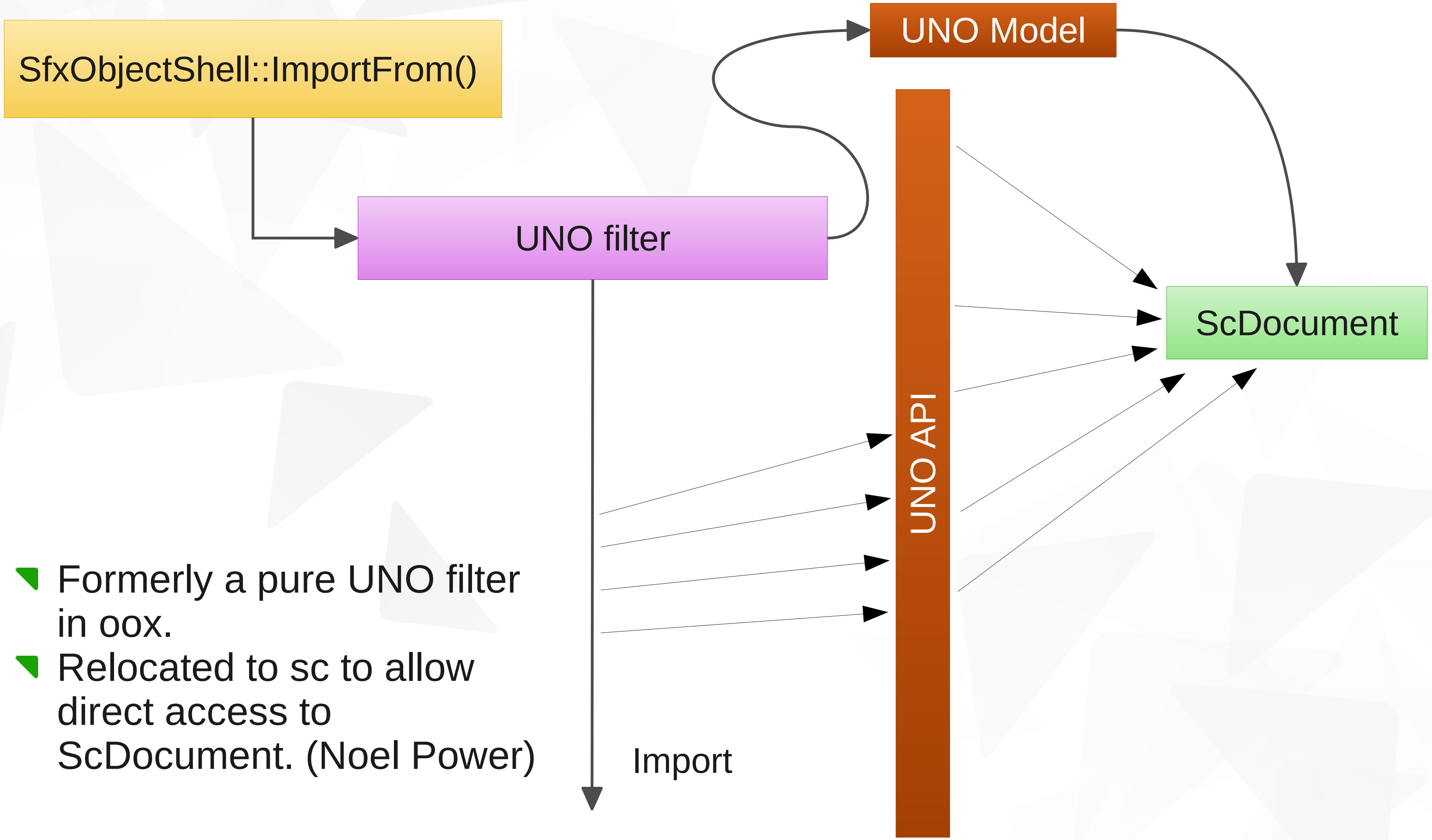
Overview (ODS import filter)


SfxObjectShell::LoadOwnFormat()
ScDocShell::Load()
UNO storage
ScDocShell::LoadXML()
ScXMLImportWrapper::Import()
UNO XML parser
UNO XML handler (ScXMLImport)
ScXMLImportWrapper::ImportFromComponent()
UNO API
UNO doc model
ScDocument

Populate document via UNO API or direct access to ScDocument.

Called per each XML stream.
(meta.xml, settings.xml, styles.xml, content.xml)

Overview (XLSX import filter)



- Formerly a pure UNO filter in oox.
- Relocated to sc to allow direct access to ScDocument. (Noel Power)

Good & Bad

- **The Good**
  - It works today, with lots of features.
  - Performance reasonable with internal filters.
  - Loaded on-demand.
- **The Bad**
  - Horrible performance with UNO API.
  - Over-complicated design. Mixture of UNO and internal APIs.
  - Not reusable outside LibreOffice.
  - No independent unit test framework.

Orcus

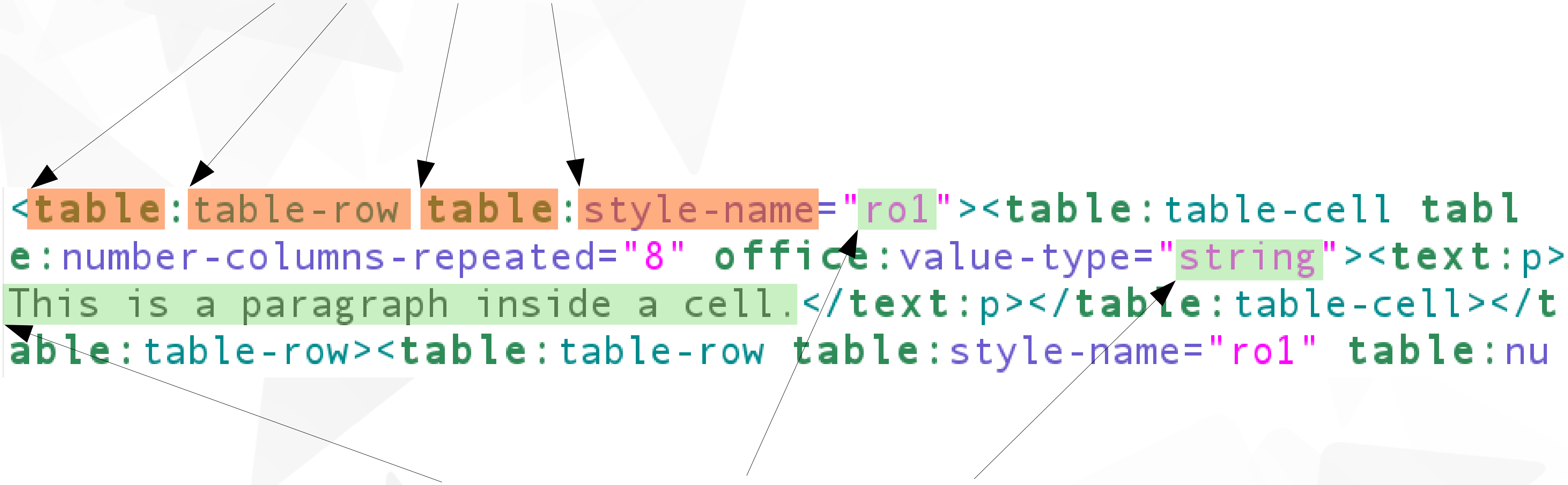
- https://gitorious.org/orcus
- Performance and maintainability focus.
- Standalone C++ library.
- Usable outside of LibreOffice.
- Two Layers
  - Base raw stream parsers (C++ templates) - XML, CSS, CSV. No linking necessary.
  - Import filters - ODS, XLSX, CSV, Gnumeric, generic XML. Gnumeric by Markus Mohrhard. Not feature-complete.
- Independent unit test framework.
- Depends on boost, zlib, libzip, mdds, ixion.

Orcus - Performance bits

- No temporary string allocations; re-use stream buffer.
- Tokenized XML parsing – avoid string comparisons.
- C++ template based parser – allow compiler optimization.
- Interface API designed for performance.
  - No temporary strings. Pass pointer to first char and length.
  - Push contents to the model while parsing. Avoid intermediate storage.

Orcus - Re-use stream buffer (XML, CSV)

Tokenized to numeric IDs.

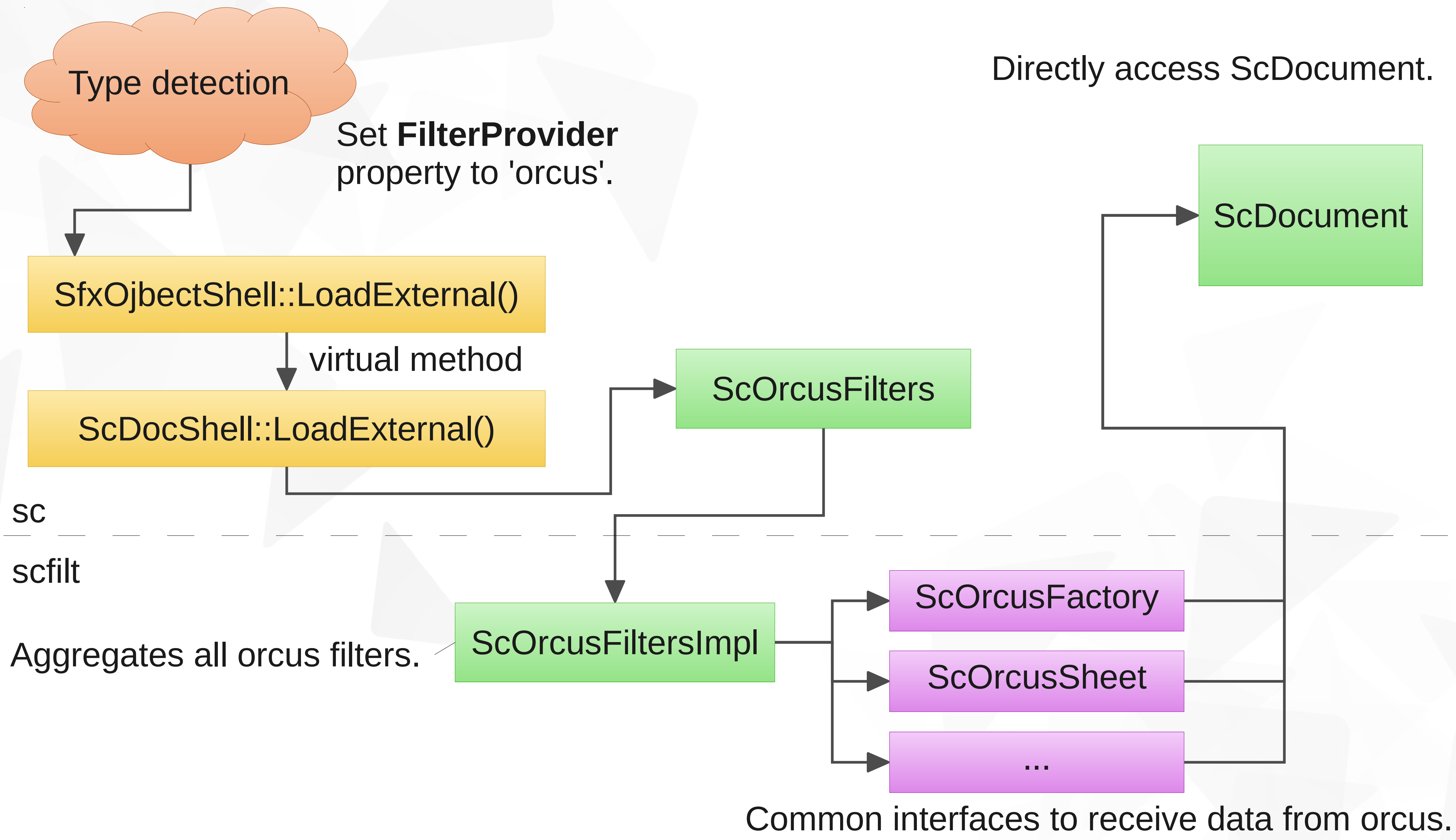


```
<table:table-row table:style-name="ro1"><table:table-cell tabl
e:number-columns-repeated="8" office:value-type="string"><text:p>
This is a paragraph inside a cell.</text:p></table:table-cell></t
able:table-row><table:table-row table:style-name="ro1" table:nu
```

Only memory address and size are stored - no allocation.
Valid while the XML buffer is in memory.

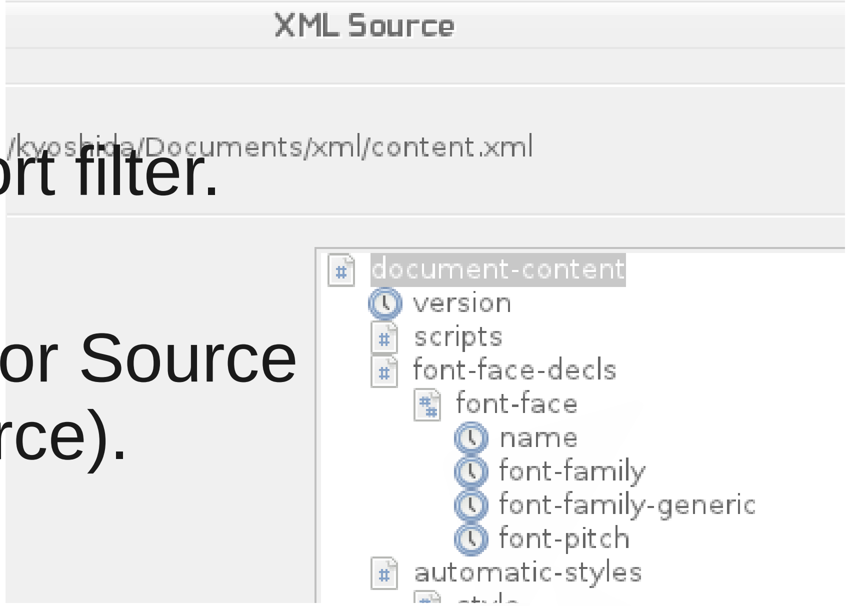
* One shared buffer for values that need conversion – XML's encoded chars, CSV's double quotes.

Orcus – How it integrates into LibreOffice


Type detection
Directly access ScDocument.
Set **FilterProvider** property to 'orcus'.
ScDocument
SfxOjbectShell::LoadExternal()
virtual method
ScOrcusFilters
ScDocShell::LoadExternal()
sc
scfilt
ScOrcusFactory
Aggregates all orcus filters.
ScOrcusFiltersImpl
ScOrcusSheet
...
Common interfaces to receive data from orcus.

Orcus - Today, and to the future.

- **At present**
  - CSS stream parser for HTML import filter.
  - CSV stream parser in filters tests.
  - Generic XML import/export filters for Source XML feature (feature/calc-xml-source).
- **In near future**
  - ODS styles import.
  - Gnumeric, CSV (need type detection support).
  - Separate ScDocument import-only API.
- **In distant future**
  - ODS, XLSX - configuration vs current filters.



XML Source

/kyoshida/Documents/xml/content.xml

Putting These All Together

Short- to mid-term plan

- **ScDocument rework**
  - Cell storage, external reference cache storage.
  - Import-only API (all filters), remove **bImportingXML**, **bLoadingMedium**, **bInsertingFromOtherDoc** etc.
- **Formula engine improvement**
  - Ixion low priority for now.
  - Shared formula
  - Range-based dependency tracking.
  - ScInterpreter to use new cell storage.
- **Orcus library**
  - More unit tests.
  - Type detection support (for Gnumeric import).
  - Use ScDocument's import-only API.

BERLIN 2012
CONFERENCE

17th-19th October

Thank You