State of the Toolchain

FOSDEM 2023

Stephan Bergmann
Red Hat, Inc.
Still at C++17

- ...almost
  - Most notably missing std::from_chars
    - Only in libstdc++ 8 (11 for fp)
    - Had to fend off rtl_str_toInt32_WithLength thrice
...and Beyond

- Forward support of some small features
  - C++20 o3tl::span
  - C++20 o3tl::cmp_equal etc.
  - C++23 O3TL_UNREACHABLE
...and Beyond

- Conditional HAVE_CPP_CONSTEVAL
  - Used in three places, to enforce assert fail at compile time
    ```cpp
    consteval Color(sal_uInt32 nColor) { assert(nColor <= 0xffffffff); }
    ```
  - But configure.ac needs to check for 2 Clang (since fixed), 2 GCC (still open), 1 MSVC (still open) bugs
...and Beyond?

- Pervasive big features:
  - C++20 Concepts
    - One conditional use in unotools
      ```cpp
      rtl::Reference<interface_type> SAL_CALL WeakReference::get() const
      requires(!cppu::detail::isUnoInterfaceType<interface_type>);
      ```
    - Would be nice to use template<Concept T> instead of std::enable_if hacks
  - C++20 Modules
    - Let's wait for more real-world experience first?
Future-Proof

- Opt-in --with-latest-c++
  - Upcoming C++23
- My test matrix:
  - Clang 17 trunk: Linux, macOS, Windows
  - GCC 13 trunk: Linux
  - Latest MSVC 2022 Preview: Windows
  - libc++ 17 trunk: with Clang on Linux, macOS
  - libstdc++ 13 trunk: with Clang, GCC on Linux