VICTOR

Interns: Victor Valov, Arthur Hemery, George Leotescu, Shamil Garifullin

Supervisor: Dimitrios Staikos

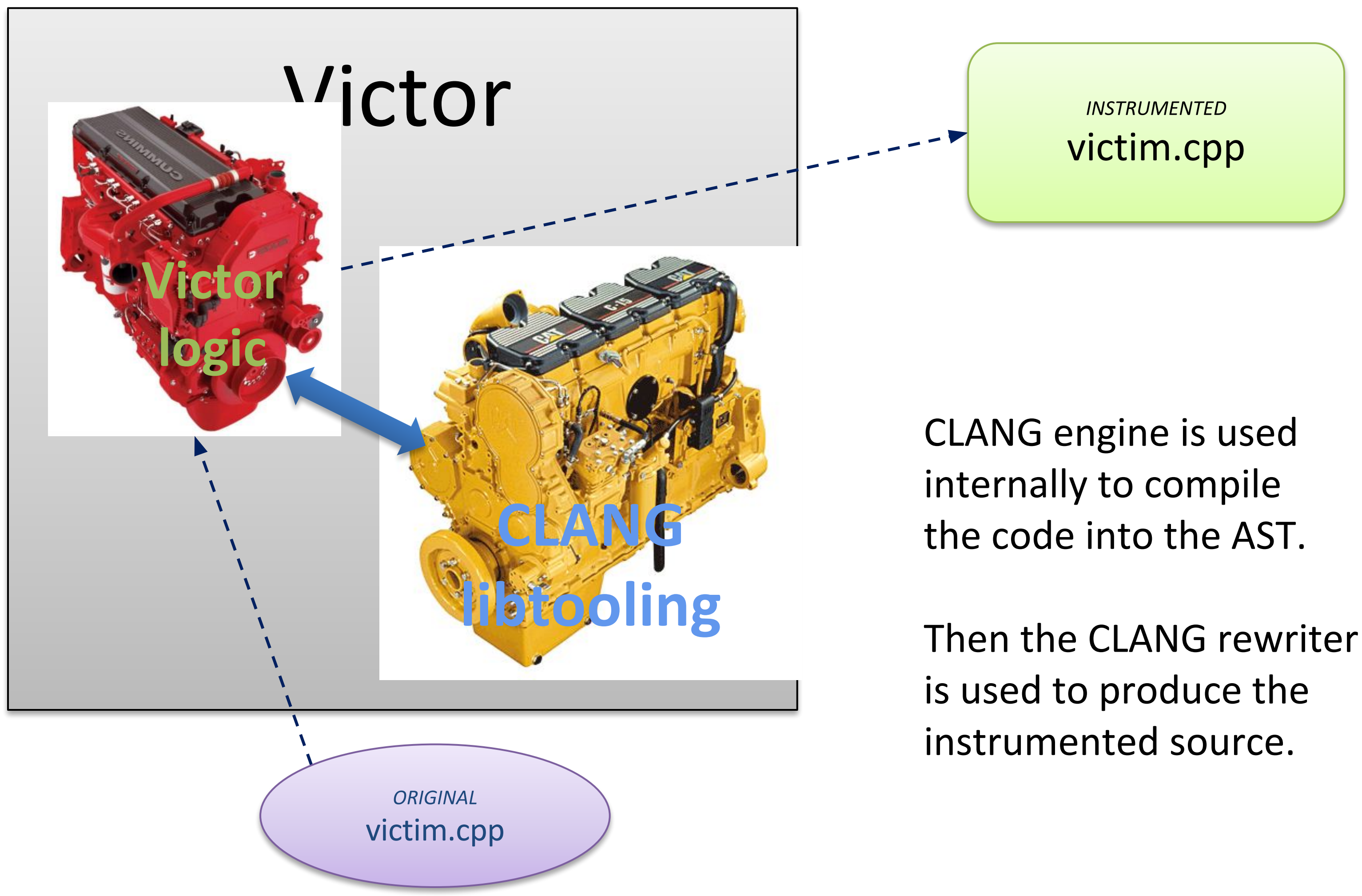
Motivation

- Increased Debugging and Troubleshooting abilities in PROD
- Minimum overhead
- No extra dependencies on tools

Simplified overview

- Inject an object at the start of functions (instrument code)
- Constructor and destructor write logs
- Postprocessor works with the output (function location, visualization, user interaction)
- Supports C and C++

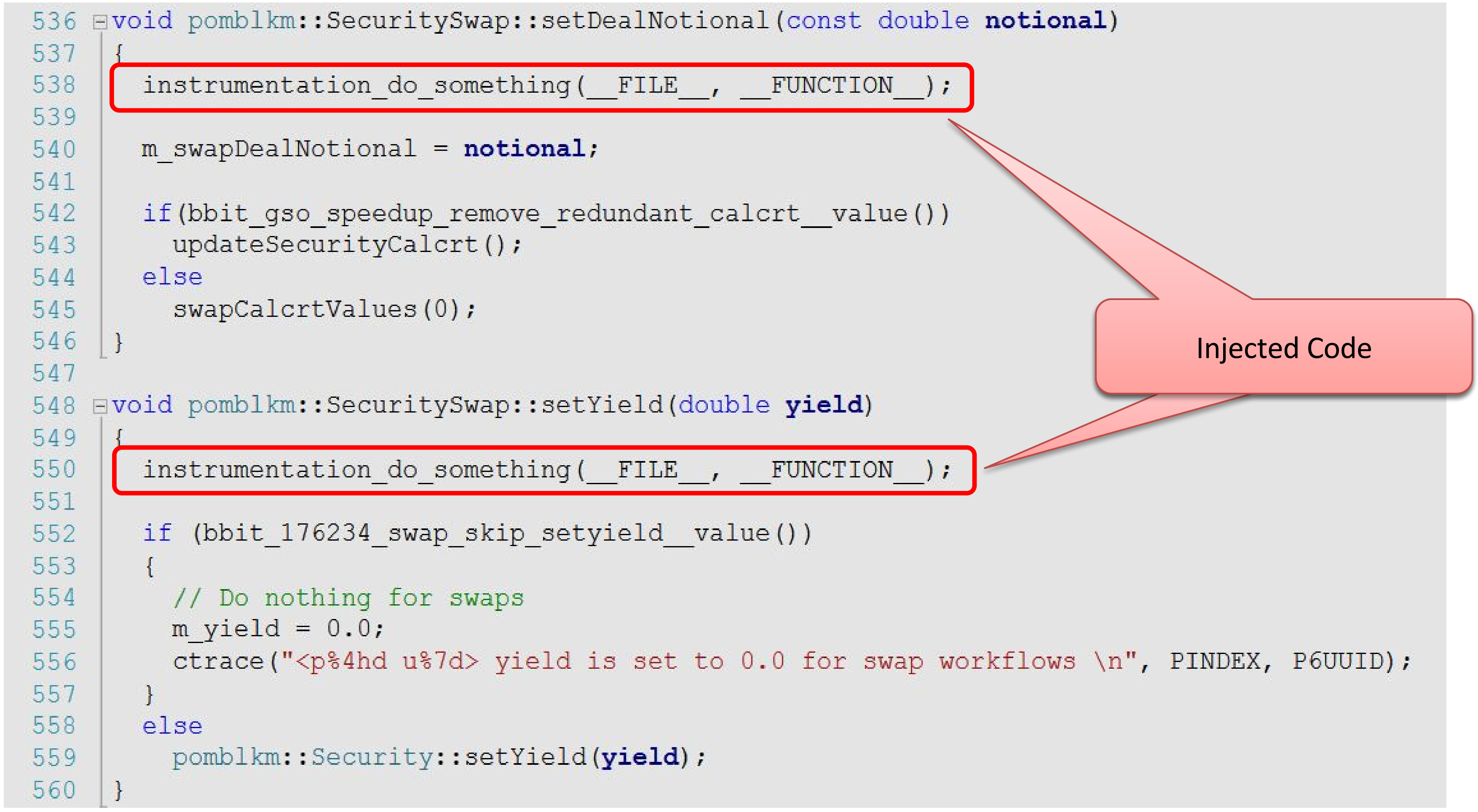
What’s inside? CLANG!



CLANG engine is used internally to compile the code into the AST.

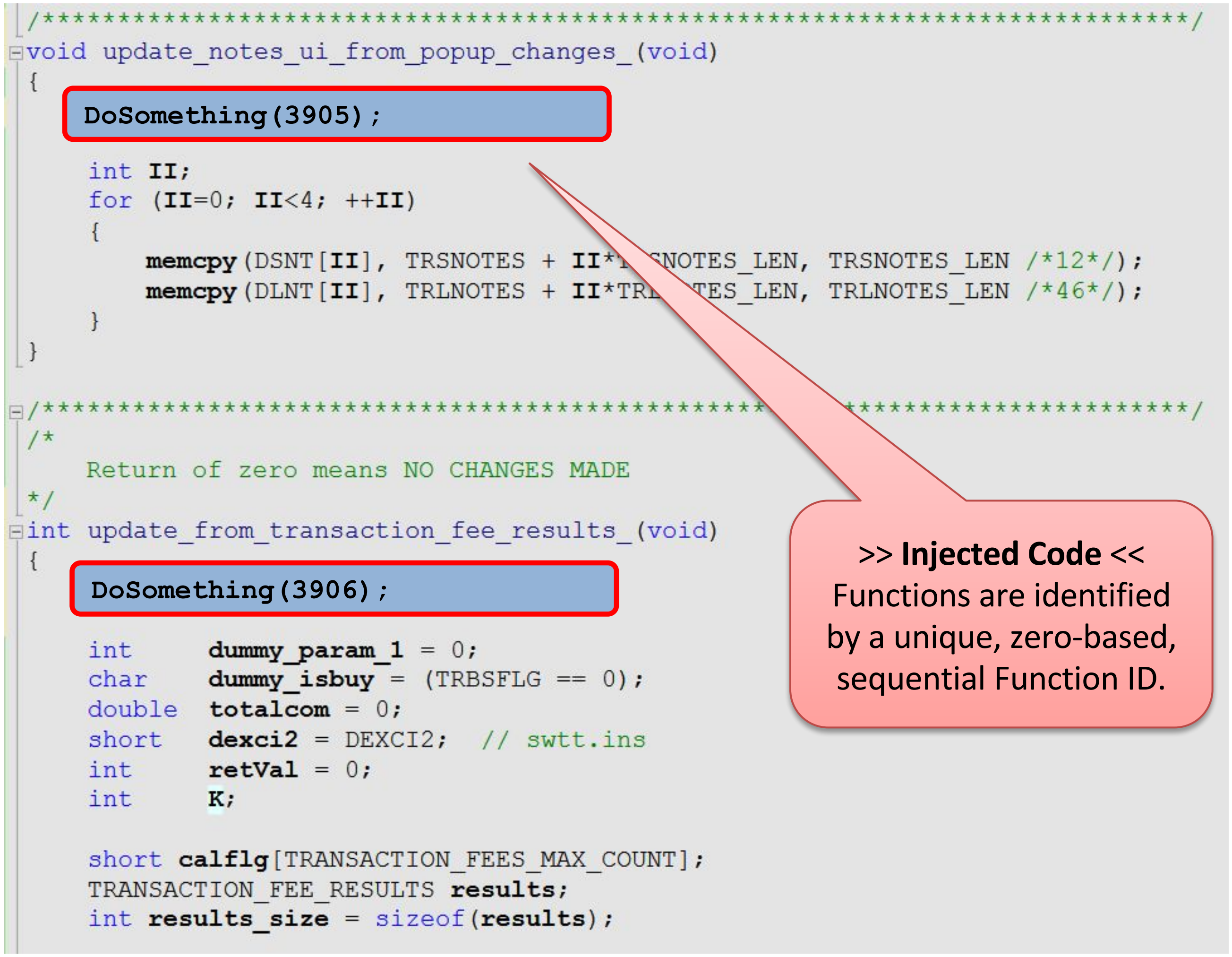
Then the CLANG rewriter is used to produce the instrumented source.

String IDs



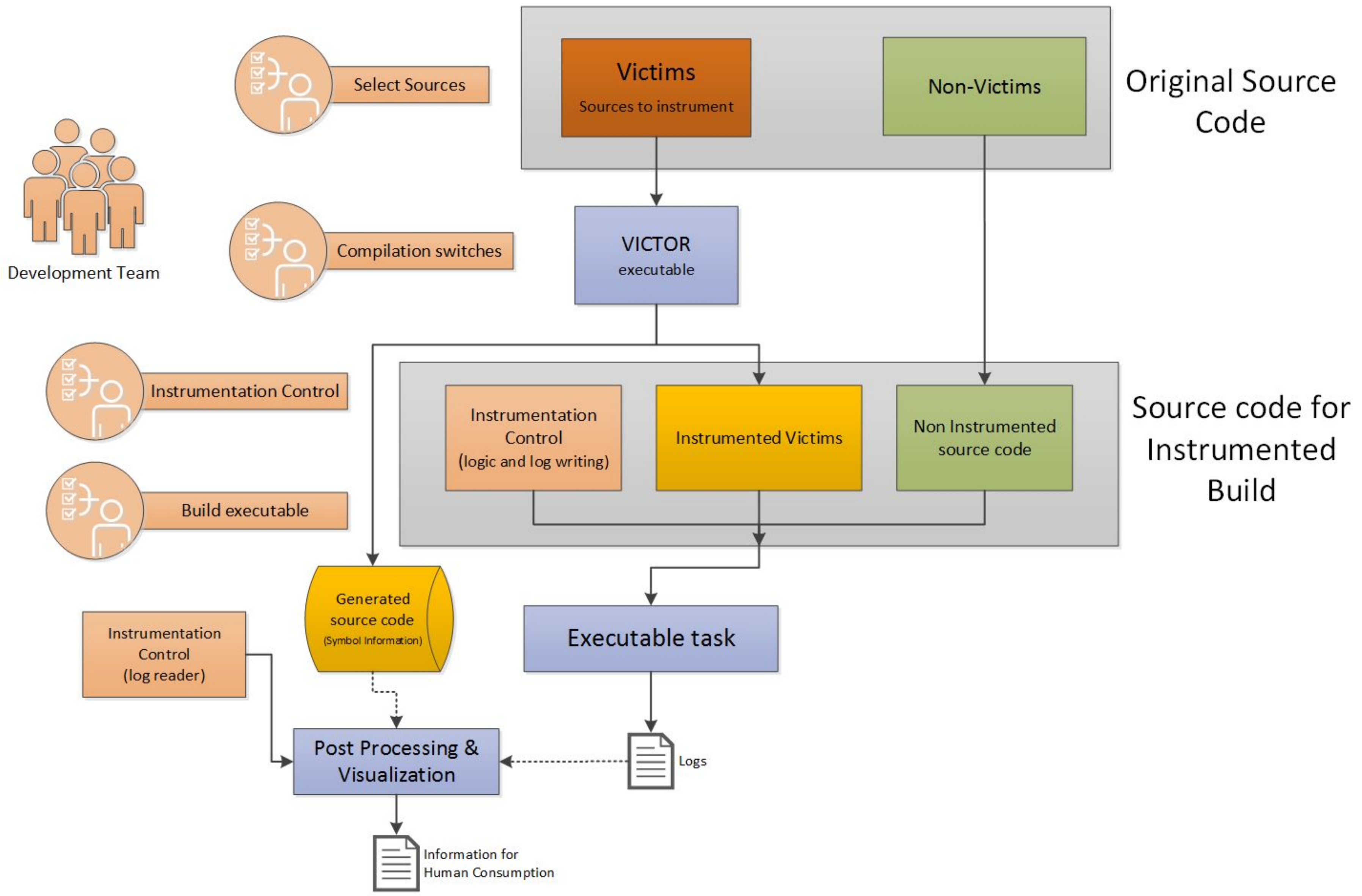
```
void pomblkm::SecuritySwap::setDealNotional(const double notional)
{
  instrumentation_do_something(__FILE__, __FUNCTION__);

  m_swapDealNotional = notional;

  if(bbit_gso_speedup_remove_redundant_calcrt__value())
    updateSecurityCalcrt();
  else
    swapCalcrtValues(0);
}

void pomblkm::SecuritySwap::setYield(double yield)
{
  instrumentation_do_something(__FILE__, __FUNCTION__);

  if (bbit_176234_swap_skip_setyield__value())
  {
    // Do nothing for swaps
    m_yield = 0.0;
    ctrace("<p%4hd u%7d> yield is set to 0.0 for swap workflows \n", PINDEX, P6UUID);
  }
  else
    pomblkm::Security::setYield(yield);
}
```

Integer IDs



```
/***************************************************************************/
void update_notes_ui_from_popup_changes_(void)
{
    DoSomething(3905);

    int II;
    for (II=0; II<4; ++II)
    {
        memcpy(DSNT[II], TRSNOTES + II*T SNOTES_LEN, TRSNOTES_LEN /*12*/);
        memcpy(DLNT[II], TRLNOTES + II*TRL TES_LEN, TRLNOTES_LEN /*46*/);
    }
}

/************************************************      ***********************/
/*
    Return of zero means NO CHANGES MADE
*/
int update_from_transaction_fee_results_(void)
{
    DoSomething(3906);

    int     dummy_param_1 = 0;
    char    dummy_isbuy = (TRBSFLG == 0);
    double  totalcom = 0;
    short   dexci2 = DEXCI2;   // swtt.ins
    int     retVal = 0;
    int     K;

    short calflg[TRANSACTION_FEES_MAX_COUNT];
    TRANSACTION_FEE_RESULTS results;
    int results_size = sizeof(results);
```

>> **Injected Code** <<
Functions are identified by a unique, zero-based, sequential Function ID.


Victims
Sources to instrument
Non-Victims
Original Source Code
Select Sources
Development Team
Compilation switches
VICTOR
executable
Instrumentation Control
Build executable
Instrumentation Control
(logic and log writing)
Instrumented Victims
Non Instrumented source code
Source code for Instrumented Build
Generated source code
(Symbol Information)
Executable task
Instrumentation Control
(log reader)
Post Processing & Visualization
Logs
Information for Human Consumption

Example Victor Applications

- Control Flow Visualization/Logging
- Remote controlled cheap-stack traces
- Instrumented failure
- …

Binary Logging Protocol

Record Types
Function Entry/Exit
Timestamp
Function Entry/Exit & Timestamp
String
Set Default Entry Type
Synch Record
EOF Record
ThreadID Record

Humanly Readable Header	Initial Header	ThreadID Record	Default Record	Record
1024 bytes	Endianness Indicator	Data	Data	Record Start: 0xFFFFFFFF
	Major Version Number			Record Type
	Minor Version Number			Data length
	Header Size			Data
	Signature of symbol DB			Record End: 0xF0F0F0F0
	Default Record Type			
	Rollover Size			
	Data Offset			

Synch Record	Synch Header (10 MB)
	Endianness Indicator
	Major Version Number
	Minor Version Number
	Header Size
	Signature of symbol DB
	Default Record Type
	Data Offset
	# Records in prev section

EOF Record	Humanly Readable Footer
	1024 bytes

Binary Logging Protocol

- The writer doesn’t allocate memory on heap
- Exception safe
- Thread aware
- The corresponding reader for post processing

Refactoring and Improvements

- Dynamic DB loader in Control Flow
- Symbol DB Signature support
- Build automation in Jenkins
- Tests: error enums, removed warnings, unhadled exceptions
- BDE isolation – conflicts with Clang (RTTI)
- Reserve zero IDs

Further Work

- Terminal integration and user interaction
- Reading of writer opened logs
- Change the build process for BPKG

Thank you!