

**The Return of the Building Blocks**  
**NeRd Talker:** Sergio Borghese

# Let's talk about...



makeameme.org

- **CachedVariable**
- **ProcessSingleton**
- **TimedCachedVariable**

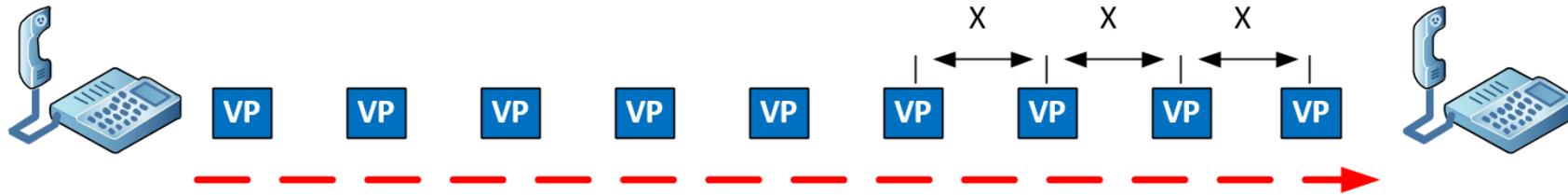


# CachedVariable

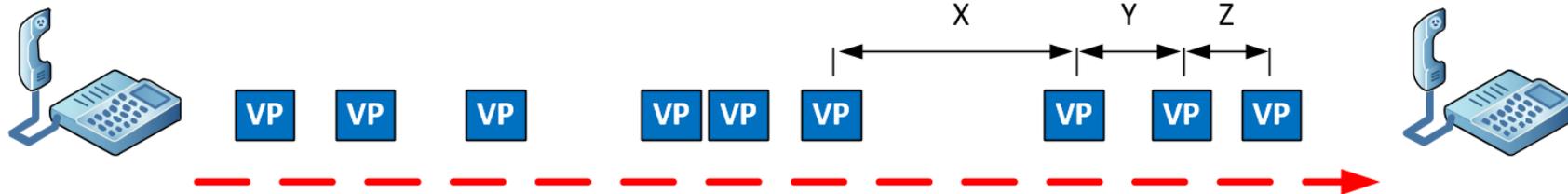
- **Class Name:** CachedVariable
- **Scope:** store a given value and it's previous value
- **Why:** simplify operations on a value and it's predecessor
- **Example:** calculate the Jitter of an RTP flow
- **Blame List:** Sergio



## No Jitter



## Jitter



*Jitter occurs when voice packets (labeled VP in the diagram) do not arrive with consistent periodicity.*

**ENOUGH BLAH BLAH**

**SHOW ME SOME CODE!**

makeameme.org

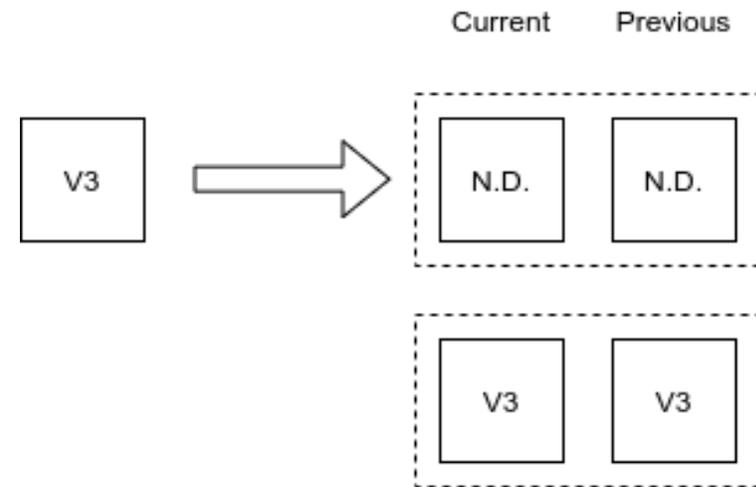
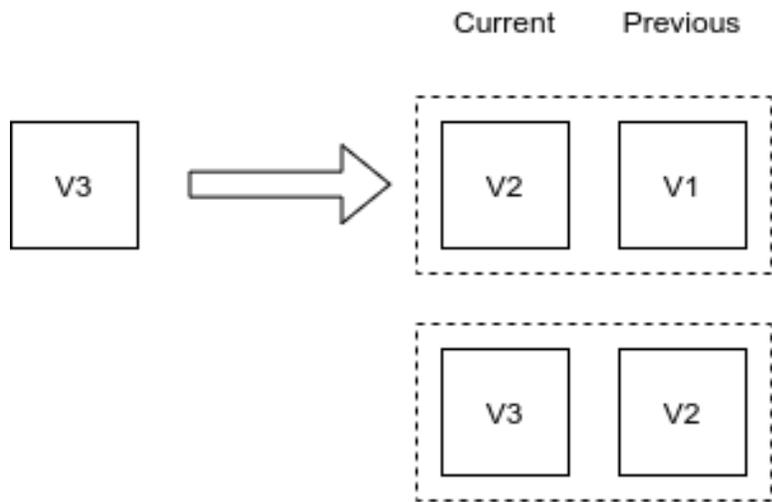


# CachedVariable Iface

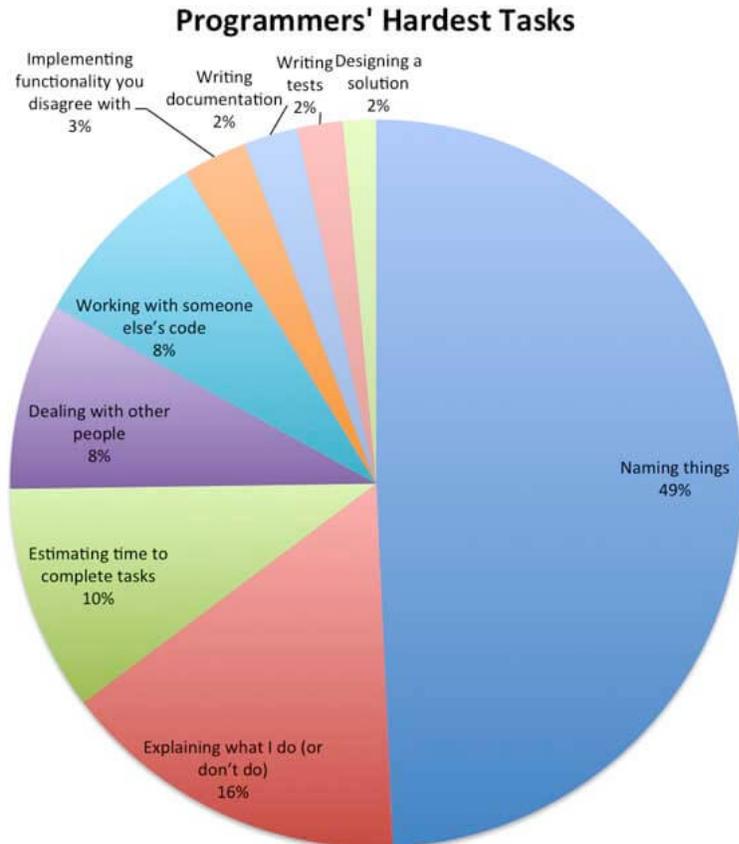
- `template<class T> class CachedVariable`
  - `CachedVariable()`
  - `CachedVariable(T aVal)`
  - `const T& operator =(const T& rhs)`
  - `bool operator ==(const T& rhs) const`
  - `bool changed() const`
  - `T current() const`
  - `T old() const`
  - `T previous() const`
  - `bool initialized() const`



`const T& operator =(const T& rhs)`



# Critique and Improvements



Data Source: Quora/Ubuntu Forums  
Total Votes: 4,522



- Name does not express the class behavior
- Need more operators, i.e. comparison
  - use with STL containers
- Extend to manage more than just 2 values ?
  - Ring buffer



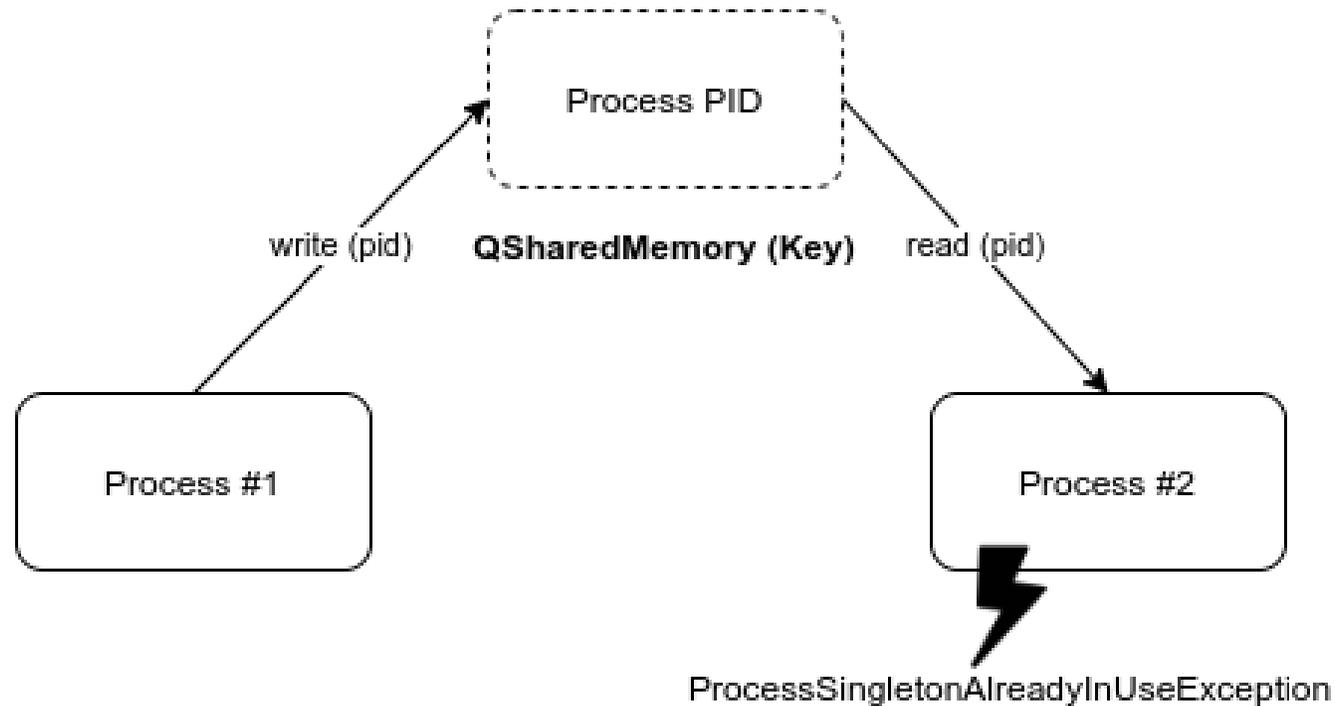
# ProcessSingleton

- **Class Name:** ProcessSingleton
- **Scope:** guarantee a single process instance is up and running on a system
- **Why:** to guarantee a single process instance is up and running on a system
- **Example:** KCTI desktop (?); Atena Bot on Appliances
- **Blame List:** Sergio & Peppe



# ProcessSingleton Iface

- `class ProcessSingleton`
  - `ProcessSingleton(const QString& aKey)`
  - `~ProcessSingleton()`



# Know your ShMem

- ShMem is handled by the Kernel
- On process **crash** the ShMem segment **IS NOT cleaned**
- Process need to detach and delete the ShMem segment explicitly
  - **PID value check is not enough**
  - **Need to check process liveness**



# Critique and Improvements

- Too much logic in the constructor?
- Too much logic in the destructor?



# TimedCachedVariable

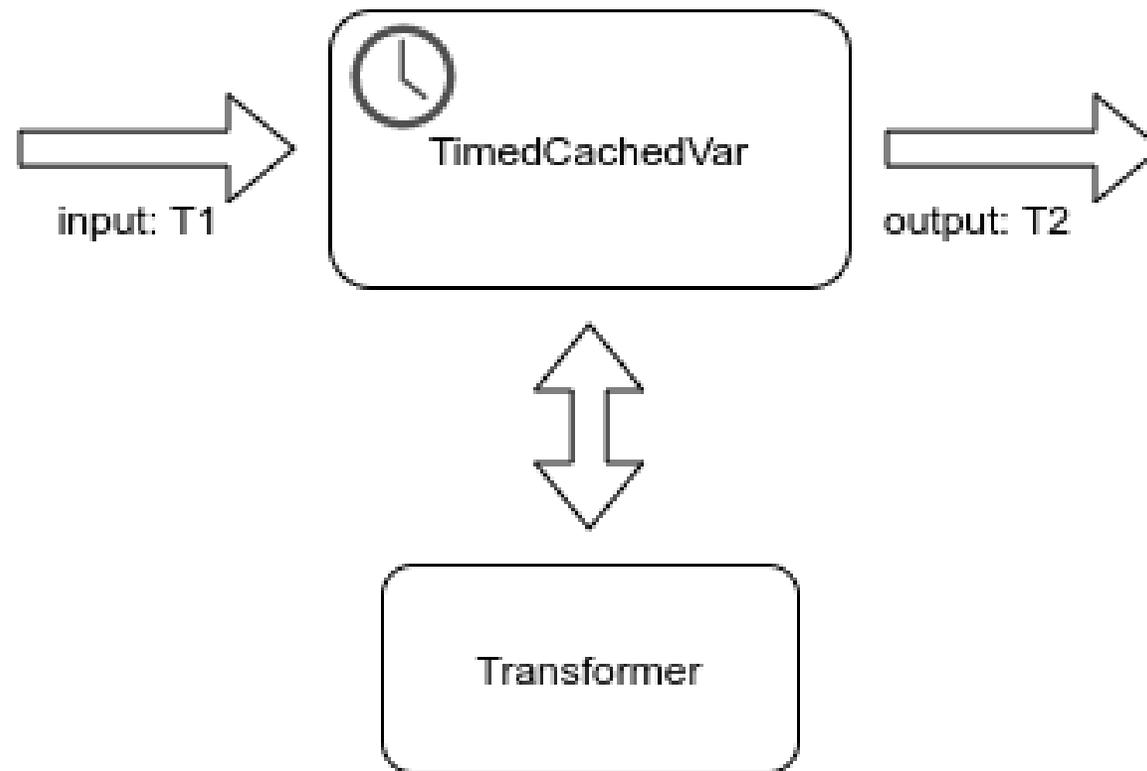
- **Class Name:** TimedCachedVariable
- **Scope:** Cache the result of a computation for a given amount of time;
- **Why:** Re-evaluate an input; but not all the times!
- **Example:** Read the ip address of an interface, should it change
- **Blame List:** Silvia & Sergio



# TimedCachedVariable Iface

- `template <class T1, class T2, class Transformer> TimeCachedVariable`
  - `TimedCachedVariable(T1 input, Transformer i_transformer)`
  - `T2 getValue()`
  - `void setCacheExpireTimeSec(qint64 i_expireTimeSec)`
  - `qint64 getCacheExpireTimeSec()`
  - `void invalidateCache()`
- Transformer **MUST** implement the API (*design by contract*)
  - `T2 transform(T1)`





# Dummy Use Case

```
class DoubleValue {  
public:  
    int transform(int x) { return x*2; }  
};  
int main () {  
    DoubleValue transformer;  
    TimedCachedVariable<int, int, DoubleValue> test(6, transformer);  
    qDebug() << test.getValue() << endl;  
    return 0;  
}
```



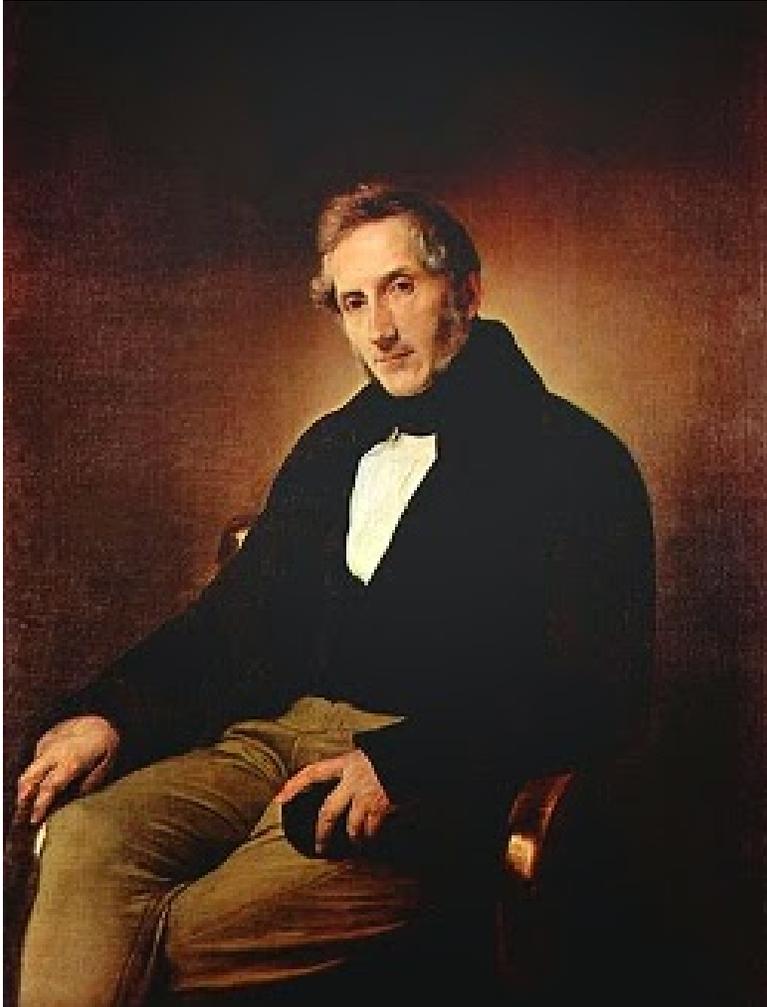
# IP Address Changing

```
class GetIfaceAddress {
public:
    QHostAddress transform(const QString & ifaceName) {
        // some complex and slow IP retrieving logic here!
        return theIpAddress;
    }
};

int foo () {
    GetIfaceAddress ipRetriever;
    TimedCachedVariable<QHostAddress, QString, GetIfaceAddress> ipToBind("eth0", ipRetriever);
    bind(ipToBind);
}
```



# Critique and Improvements



- Class name... again!?!
- RFC: Request For Comments



## References

- **CachedVar**
  - Vdk or Atena Bot Code :-/
- **TimeCachedVar**
  - [gitlab.netresults.dev:10443/netresults/utils/timecachedvar.git](https://gitlab.netresults.dev:10443/netresults/utils/timecachedvar.git)
- **ProcessSingleton**
  - [gitlab.netresults.dev:10443/netresults/utils/processsingleton.git](https://gitlab.netresults.dev:10443/netresults/utils/processsingleton.git)
- **Shared Memory**
  - [https://en.wikipedia.org/wiki/Shared\\_memory](https://en.wikipedia.org/wiki/Shared_memory)