**Архитектура**

**Протокол взаимодействия**

Приложение будет основано на шаблоне “Издатель-Подписчик” (Данная схема легко масштабируется).

Утилита управления будет рассылать на Агенты информацию (настройки, подписывается / не подписывается, …), на основе которой Агент будет рассылать своим подписчикам (т.е Утилите управления) данные. Взаимодействие между Утилитой и Агентом будет осуществляться на базе библиотеки ZeroMQ. На запросы Утилиты Агент будет возвращать код ошибки, если та будет иметь место.

**Формат файла .config**

Задать конфигурацию системы можно как на машинах Клиентов, так и на машине Администратора. На каждой машине будет присутствовать файл .config в формате XML, содержащий поля настроек и некоторую служебную информацию.

Пример для машины Клиента:

<settings>

…

<timelimit> 4000 </timelimit> //Период обновления данных в мс

…

</settings>

<subscribe>

<id> … </id> //Подписчик - машина Администратора

<where> … </where>

</subscribe>).

Работа программы с файлами .config будет осуществляться с помощью библиотеки libxml2, работа Администратора с файлами .config – с помощью командной строки (а потом и GUI оболочки), работа Клиента с файлами .config не предполагается.

**Log файлы**

В программе будет присутствовать журналирование: на машинах Клиентов и машине Администратора будут располагаться файлы log/<DATA>.log. В папке log будут содержаться файлы логов за время N в днях (настраивается в .settings) в формате:

[dd.mm.yyhh.mm.ss] <Начало сессии>

…

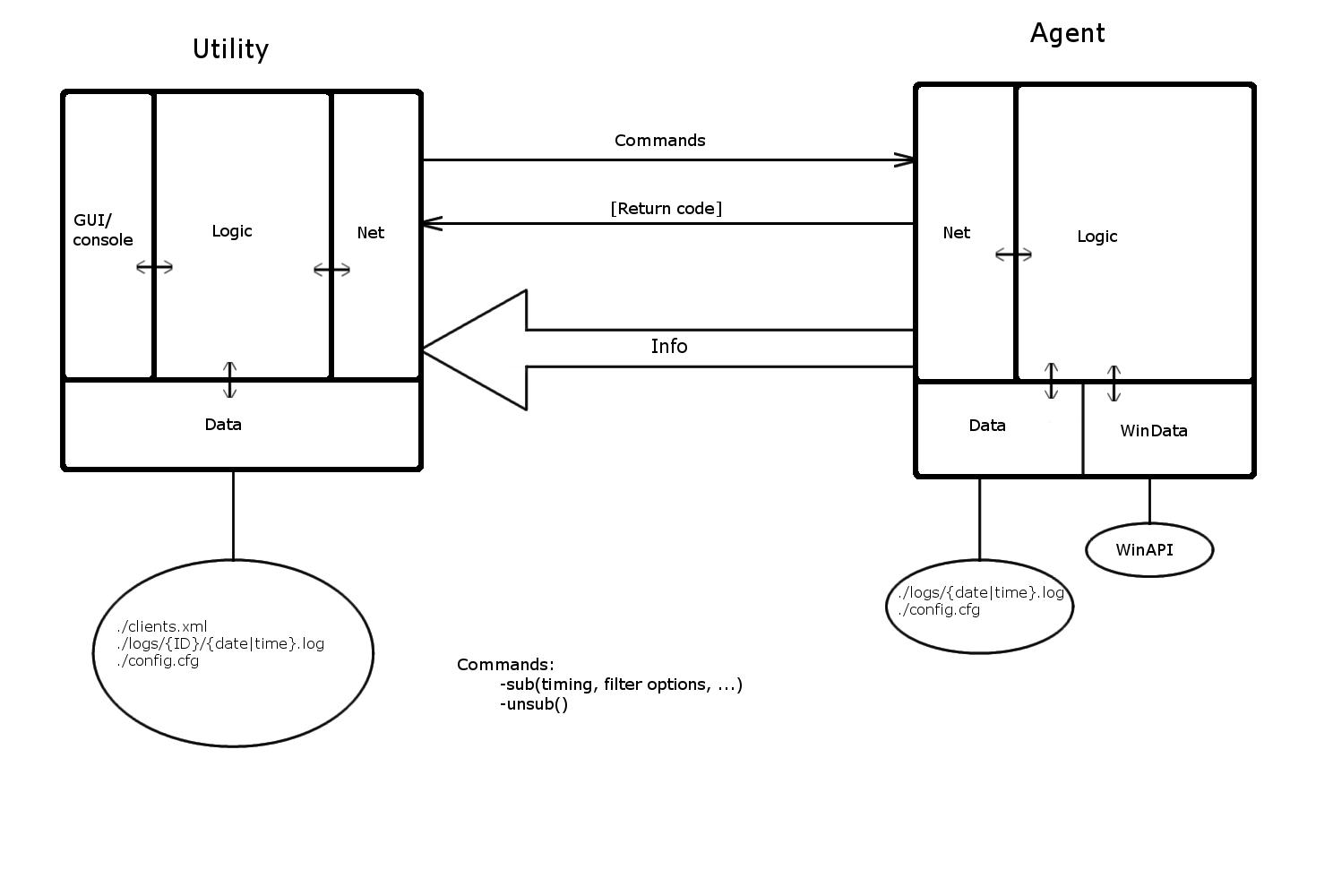
[dd.mm.yyhh.mm.ss] Было изменено время ….

[dd.mm.yyhh.mm.ss] Добавлен новый subscriber\_<ID>

…

[dd.mm.yyhh.mm.ss] <Конец сессии>

…

Формат файлов логов совпадает на машине Администратора и машинах Клиентов, работа программы с файлами логов на основе самописной библиотеки. Работа с файлами логов Клиента на машине Клиента – ‘Визуально-Тыкательная’ в блокноте. ****Работа администратора до появления GUI совпадает со схемой работы Клиента.

**Примечания:**

Восстановления после сбоя на данном этапе не предусмотрено, обработка ошибок будет осуществляться на базе возвращенного Агентом кода ошибки запроса Утилиты.

Проверки корректности данных на данном этапе не предусмотрено.

На данном этапе приложение будет небезопасно.

На данном этапе не будут осуществлены механизмы транзакции и контроля состояния.

На данном этапе не будет проводиться кэширование и оптимизация процессов.

**Описание базы данных на машине админа:**

a) Информация о каждом клиенте на данном этапе будет храниться в виде xml структуры в отдельном файле (потом, возможно, в БД). Наиболее вероятный пример структуры:

<agents>

<agent>

<id> ... </id>

<name> ... </name>

<state> active/wait/passive </state> //состояние Агента - посылает/ожидает/не посылает

<ip> ... </ip>

<port> ... </port>

<params>

<waitT> ... </waitT> //время рассылки Агентом сообщений

... //фильтр рассылки

</params>

</agent>

...

</agents>

б) Заполнение этого файла (далее БД) осуществляется как и "ручками" в файле БД (на этапе отладки), так и с помощью интерфейса командной строки Утилиты с помощью соответствующих функций [-добавить/удалить Агента].

в) Смотри п.1.а). Но скорее это будет не слияние БД с config-файлами. На машине с Утилитой будет свой config-файл, а котором будет иная информация (не БД).

г) В случае выключения компьютера Утилиты во время записи в БД - без механизма транзакций ничья целостность не гарантируется; в случае выключения Утилиты "нормальным" способом все active-агенты переходят в режим wait, для соответствующих Агентов вызывается функция [-отписаться]. При включении Утилиты снова для всех wait-агентов вызывается функция [-подписаться] со всеми параметрами, указанными в БД (механизм сессий).

**Процесс работы работы с Агентами**

а) Подключение новых и удаление старых осуществляется функциями [-добавить/удалить Агента] (1.б)).

б) Для "активации" Агента требуется для неактивного Агента вызвать [-подписаться] с соответствующими настройками периода, фильтра и т.д., на что Агент вернет Утилите "код возврата" (успешное подключение / ...).

в) Для "деактивации" Агента требуется для активного Агента вызвать [-отписаться], Агент ничего не возвращает.

г) Формат передаваемых данных Data - xml-сериализованные данные.

д) Если упадет Агент, то, возможно, по отсутствию входных данных от него Утилита должна понимать, что Агент "сломан", и реагировать на это (на первых этапах не предусмотрено).

е) У Агента блоки: net - работа с сетью, data - работа с win.api (+ logs + config), logic - связующие компоненты. У Утилиты: net - работа с сетью, data - работа с БД (+ logs + config), logic - связующие компоненты, console/gui - внешний пользовательский интерфейс.

**Журналирование**

Журналирование будет вестись на каждой машине отдельно за время указанное в config файле. При вызове [-просмотретьЛог] с соответствующими настроками ... (не актуально до появления gui).