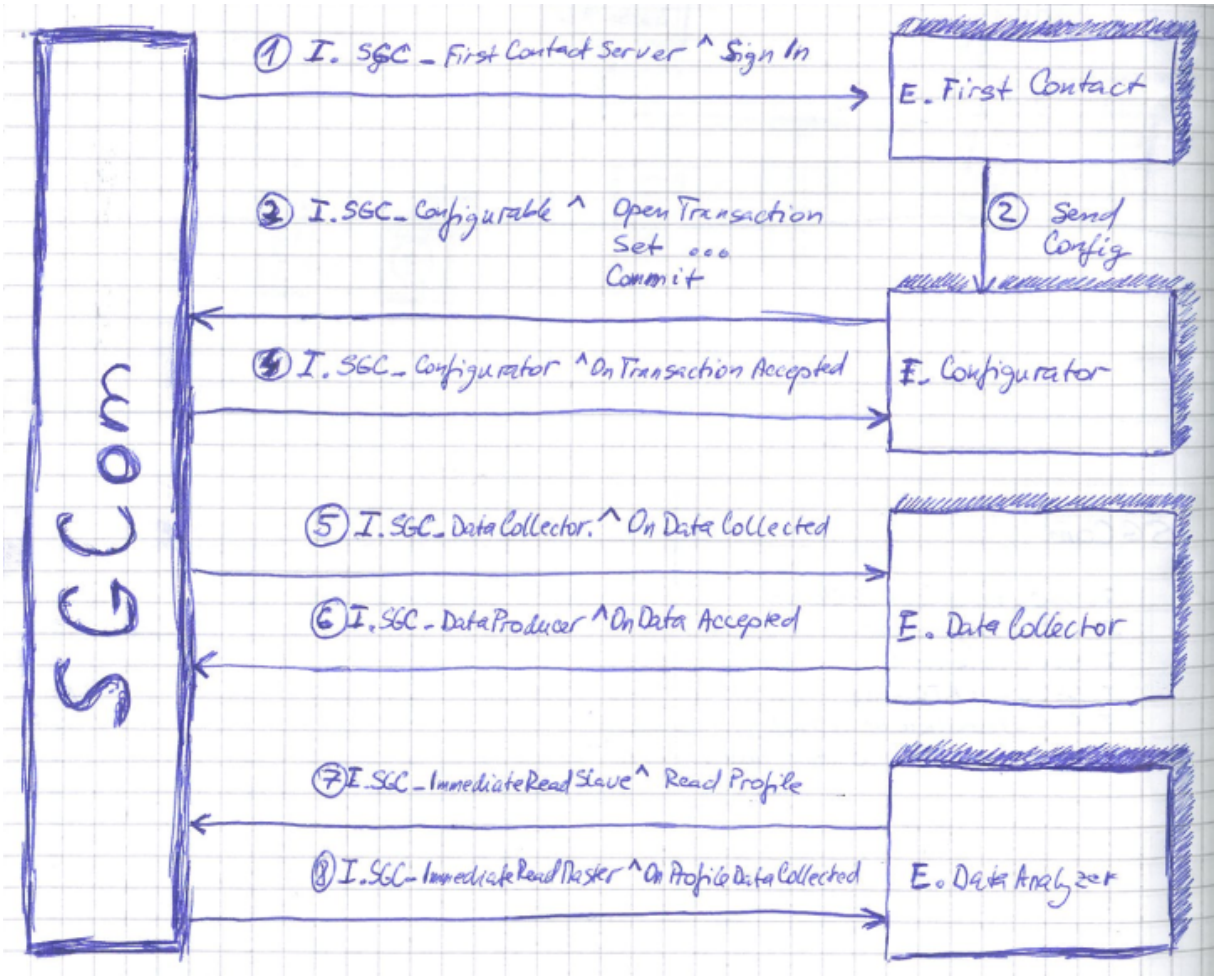
Životný cyklus SGCom-u

SGCom je navrhnutý tak, aby bol odolný voi rôznym formám výpadkov. Pri svojom štarte sa musí pripoji k centrále, aby získal konfiguráciu (bez ktorej nevykoná žiadnu akciu), ale potom môže pracova aj v prípade, že stratí s centrálou na nejaký as kontakt. Je robustný aj v prípade, že nedokáže vykona zber dát, o om informuje centrálu a poskytuje mechanizmy, ako neskôr dáta opätovne získa. Životný cyklus ilustruje nasledujúci obrázok a možno ho tiež popísa v nasledovných krokoch.



Obrázok 3 Ilustrácia životného cyklu

1. Štart aplikácie, spracovanie parametrov príkazového riadku, ktoré okrem iného obsahujú adresu pre pripojenie sa ku D2000 kernelu, prihlasovacie údaje a adresu tzv. „bodu prvého kontaktu".
2. Vykonanie údržby modulu *DataStorage*:
   - Vymazanie záznamov starších ako 90 dní. Úloha sa opakuje každých 24 hodín.
   - Odstránenie oznaenia íslom transakcie zo všetkých záznamov (podrobnosti v kroku 7).
3. Pripojenie ku D2000 kernelu centrály. SGCom sa pokúša získa D2Japi Session, ktorá bude v DODM reprezentovaná objektom s názvom v tvare SGCOM1.SGC. V prípade neúspechu sa pokus o pripojenie opakuje každých 30 sekúnd.
4. Prihlásenie sa do „bodu prvého kontaktu" Adresa objektu je definovaná parametrom z príkazového riadku –f., o je inštancia objektu typu D2000 Event, ktorá implementuje rozhranie *I.SGC_FirstContactServer*. Prihlási sa volaním RPC *SignIn* s hodnotou parametra *_name* nastavenou na meno procesu (z príkazového riadku). RPC je asynchrónne a zakladá aplikane riadenú transakciu, ktorá zabezpeí, že sa bu SGCom alebo centrála dozvie, ke jej partner havaruje.
5. Centrála na nadviazanie prvého kontaktu reaguje tým, že odošle do SGCom-u novú konfiguráciu, resp. zmení existujúci. D2000 Event, ktorý riadi zasielanie konfigurácie musí implementova rozhranie *I.SGC_Configurator*, SGCom pre prijatie implementuje rozhranie *I.SGC_Configurable*.
   a. Prenos konfigurácie je transakný. Všetky zmeny sú bu v poriadku a prijaté ako celok, alebo sú všetky zamietnuté a konfigurácia je ponechaná v pôvodnom stave. Postupnos krokov je nasledovná:
      i. Otvorenie transakcie: *I.SGC_Configurable^OpenTransaction*
      ii. Odoslanie zmien: *I.SGC_Configurable^Set** a *Delete**.
   b. Príkazy je nutné zadáva v takom poradí, aby nebola v žiadnom okamihu porušená referenná integrita konfiguranej databázy.
      i. Commit transakcie: *I.SGC_Configurable^Commit*
      ii. Potvrdenie prijatia zmien *I.SGC_Configurator^OnTransactionAccepted*, alebo zamietnutie zmien *I.SGC_Configurator^OnTransactionDenied* so zoznamom chýb v parametri.
   c. Alternatívne môže by transakcia zrušená zo strany centrály volaním *I.SGC_Configurable^Rollback*, o SGCom potvrdí *I.SGC_Configurator^OnTransactionRolledBack*. Transakcia je automaticky zrušená aj v prípade, že sa preruší aplikane riadená konverzácia.

6. SGCom poda aktuálnej konfigurácie SGCom nastaví svoj harmonogram úloh – pre každú entitu *PeriodicEvent* vytvorí zvláš záznam a naplánuje jej vykonanie.
7. Ke nastane asový okamih, na ktorý je naplánovaná periodická udalos, SGCom prezrie aktuálnu konfiguráciu a zane vykonáva všetky súvisiace úlohy. Do modulu *TaskExecutor* vloží do príslušných front úlohy na zber a odovzdávanie dát, ktoré *TaskExecutor* následne vykoná. Vykonanie úloh, ktoré si vzájomne nekonkurujú je paralelné. (Pozn.: vzájomne si konkurujú dve úlohy, ktoré sú vykonané na tom istom logickom zariadení a tiež úlohy, ktoré obsluhujú logické zariadenia pripojené tou istou komunikanou linkou. Zber a odovzdávanie dát si vzájomne nekonkuruje.)
   Zber dát každej jednej veliiny, i už úspešný alebo neúspešný, koní tým, že je výsledok zaznamenaný do modulu *DataStorage*. Pri úspešnom zbere je to zoznam hodnôt s asovými znakami, pri neúspešnom je to kód a popis chyby, tiež s asovou znakou.
   Odovzdanie hodnôt centrále prebieha po jednotlivých logických zariadeniach a jednotlivých veliinách (DataPoint). Každé odovzdanie predstavuje samostatnú transakciu a až ke centrála potvrdí úspešné prevzatie hodnôt, sú dáta odstránené z *DataStorage*. Odovzdanie pre jedno logické zariadenie a veliinu pozostáva z nasledujúcich krokov:
   a. Vygenerovanie unikátneho ID transakcie.
   b. Všetky záznamy pre dané logické zariadenie a veliinu sú oznaené íslom transakcie. (V prípade, že predchádzajúca transakcia neskonila, sú oznaené len nové záznamy.)
   c. Hodnoty a asové znaky oznaených záznamov sú odoslané volaním *I.SGC_DataCollector^OnDataCollected*.
   d. Po spracovaní dát je transakcia ukonená volaním *I.SGC_DataProducer^OnDataAccepted*.
   e. Záznamy oznaené príslušným íslom transakcie sú vymazané z *DataSorage*-u.
      (Pozn.: RPC komunikácia prebieha v rámci aplikane riadenej transakcie, takže ak spracovanie odovzdaných dát z nejakého dôvodu zlyhá, SGCom sa túto skutonos dozvie a transakcia sa zruší. Dáta zostanú v *DataStorage*-i do nasledujúceho plánovaného odovzdávania dát a oznaenie íslom transakcie sa odstráni.)
8. Centrála môže kedykovek po odoslaní konfigurácie zada príkaz na vykonanie „okamžitej" úlohy. Najastejšie sa vykonáva ítanie archivovaných dát meraných veliín za obdobie, pre ktoré v centrále chýbajú dáta, ale tiež nastavenie presného asu v meraoch a iné. Príkazy sú riadené párovými rozhraniami *I.SGC_ImmediateReadMaster – Slave*, *I.SGC_SetRTCMaster – Slave*, *I.SGC_ConsumerDisconnectMaster – Slave*, kde centrála implementuje *Master* rozhranie a SGCom implementuje *Slave*. Všetky príkazy prebiehajú v režime aplikane riadených konverzácií ale vždy pozostávajú len s jednoduchej výmeny príkaz – oznámenie výsledku.
9. Ak SGCom stratí spojenie s centrálou, naalej pokrauje vo vykonávaní plánovaných úloh, ako je popísané v bode 7. Zber dát prebieha bez zmeny, všetky pokusy o odovzdanie zlyhajú a zozbierané dáta sa preto zhromažujú v *DataStorage*-i až dovtedy, kým sa nepodarí spojenie obnovi a dáta odovzda. Po nadviazaní nového spojenia pokrauje vykonávanie bodom 4.

Modul *TaskExecutor*

Modul *TaskExecutor* je zodpovedný za vykonávanie všetkých úloh, ktoré vyplývajú z konfigurácie ako aj všetkých „okamžitých" úloh. Používa pri tom zoznam pravidiel, ktoré urujú, ktoré úlohy je možné vykona paralelne. Úlohám, ktoré paralelne vykona nemožno (lebo si vzájomne konkurujú v prístupe k meraom), definuje poradie, v akom sa vykonajú tak, aby bola maximalizovaná priepustnos. Zárove tieto pravidlá definujú asové okná, poas ktorých komunikácia s niektorými merami nie je možná – po ukonení spojenia nie je možné uritý as nadviaza nové spojenie.
Pri vytváraní konfigurácie je potrebné bra do úvahy tieto pravidlá, ako aj as, ktorý trvá vytvorenie spojenia, vykonanie jednotlivých úloh, zatvorenie spojenia a výluka spojenia, aby nebolo naplánovaných viac úloh, ako je možné obslúži.
Pri obsluhe meraov sa vykonávanie úloh riadi nasledujúcimi pravidlami:

1. Nie je možné vytvori súasne dve alebo viac spojení s jedným logickým zariadením.
2. Nie je možné vytvori súasne spojenia na dve logické zariadenia, ktoré sú pripojené tou istou komunikanou linkou (v konfigurácii zdieajú tú istú entitu *Connector*).
3. Ak existuje aktívne spojenie k nejakému logickému zariadeniu, musia sa vykona všetky úlohy, ktoré sú pre dané zariadenie zaradené do fronty úloh skôr, ako bude spojenie ukonené (z optimalizaných dôvodov). Úlohy sú z fronty vyberané politikou „first-came-first-served".
4. Po vykonaní poslednej úlohy z fronty zostáva spojenie k logickému zariadeniu aktívne ešte 10 sekúnd Džku akania na novú úlohu je možné upravi parametrom z príkazového riadku -n.. Ak je v tomto asovom intervale do fronty zaradená nová úloha, je tiež vykonaná a spojenie sa opä ponecháva aktívne alších 10 sekúnd. Až ke nepribudne žiadna nový úloha, je spojenie uzatvorené. (Aby bolo možné efektívne obslúži rýchlo sa opakujúce úlohy.)
5. Po zrušení spojenia nie je možné vytvori nové spojenie na žiadne z logických zariadení, ktoré sú pripojené použitou komunikanou linkou po dobu 20 sekúnd Džku ochranného intervalu je možné zmeni parametrom z príkazového riadku -s.. (Lebo by bolo spojenie aj tak odmietnuté a trvalo by to dlhšie, kým by sa spojenie podarilo opä nadviaza.)
6. Po uplynutí 20-sekundového ochranného intervalu je obslúžené alšie logické zariadenie akajúce vo fronte.
7. Ak je pokus o vytvorenie spojenia s nejakým logickým zariadením neúspešný 3-krát po sebe Poet pokusov je možné zmeni parametrom z príkazového riadku -r., považuje sa za odpojené – stav *HardError* – a spojenie s ním nie je možné nadviaza po dobu nasledujúcich 5 minút Džku trvania *HardError* stavu je možné upravi parametrom z príkazového riadku -d.. Všetky úlohy vo fronte, ktoré mali by vykonané na tomto zariadení, sú stornované, ako aj všetky nové úlohy po dobu trvania *HardError* stavu. Po uplynutí tejto lehoty je možné sa opä pokúsi pripoji k zariadeniu.
8. Všetky úlohy, ktoré si vzájomne nekonkurujú, je možné vykona paralelne vláknami príslušného *ThreadPool*-u. Poet vlákien v *ThreadPool*-e je maximálne 512 Maximálny poet vlákien je možné upravi parametrom z príkazového riadku -t., ale dynamicky sa mení poda aktuálnych potrieb.