



Regione Toscana



Architettura Generale

D2.8.2

Versione 1.2
Data: 17-01-2014



Progetto iCaro

La piattaforma cloud per l'accelerazione
del business delle PMI toscane
[CUP 6408.30122011.026000074]

COMPUTER
GROSS


liberologico.com



UNIVERSITÀ
DEGLI STUDI
FIRENZE
DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE



 **altro
lavoro**
Agenzia per il lavoro



Informazioni sul documento

ID Deliverable	D2.8.2
Titolo Deliverable	Architettura Generale
ID Attività	2.1
N. Versione / Revisione	1.2
Natura: Bozza / Definitivo	Definitivo
Partner responsabile	UNIFI
Distribuzione: Riservato / Pubblico	Riservato
Riferimenti Autore	Paolo Nesi, paolo.nesi@unifi.it , Daniele Cenni, daniele.cenni@unifi.it
Data redazione	17/01/2014
Riferimenti revisore	Tutti i partner
Data revisione	17/01/2014
Riferimenti soggetto che approva	ICARO consorzio
Data approvazione e consegna	17/01/2014

Controllo delle revisioni

Oggetto	Numero	Data
Prima stesura	1.0	17/12/2013
Versione finale	1.1	17/01/2014

Nota di riservatezza

Il presente documento sarà utilizzato esclusivamente ai fini del progetto ICARO, ha carattere riservato e non potrà quindi essere divulgato se non in seguito ad esplicita autorizzazione scritta da parte dell'ATS, salvo il caso in cui di richieste di ottemperare ad obblighi di legge o a richieste di pubbliche autorità.





Indice

1. Aspetti Generali come da progetto.....	4
2. Architettura Generale.....	6
2.1 Subscription Portal (SP).....	8
2.2 Configuration Manager (CM).....	9
2.3 Business Producer (BP).....	10
2.4 Supervisor and Monitor (SM).....	11
2.5 Smart Cloud (SC).....	13
2.6 Cloud Middleware (CMW).....	14

Legenda Acronimi e sigle

Acronimo / Sigla	Dettaglio
BP	Business Producer
BPaaS	Business Process as a Service
CMW	Cloud Middleware
CRM	Customer Relationship Management
IaaS	Infrastructure as a Service
KB	Knowledge Base
MaaS	Middleware as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
SC	Smart Cloud
SCE	Smart Cloud Engine
SDK	Software Development Kit
SM	Supervisor and Monitoring
SP	Service Portal
WP	Wrapper



1. Aspetti Generali come da progetto

In questa sezione viene descritta l'architettura del sistema BPaaS di **iCaro**. L'architettura di massima è descritta nella prossima figura.

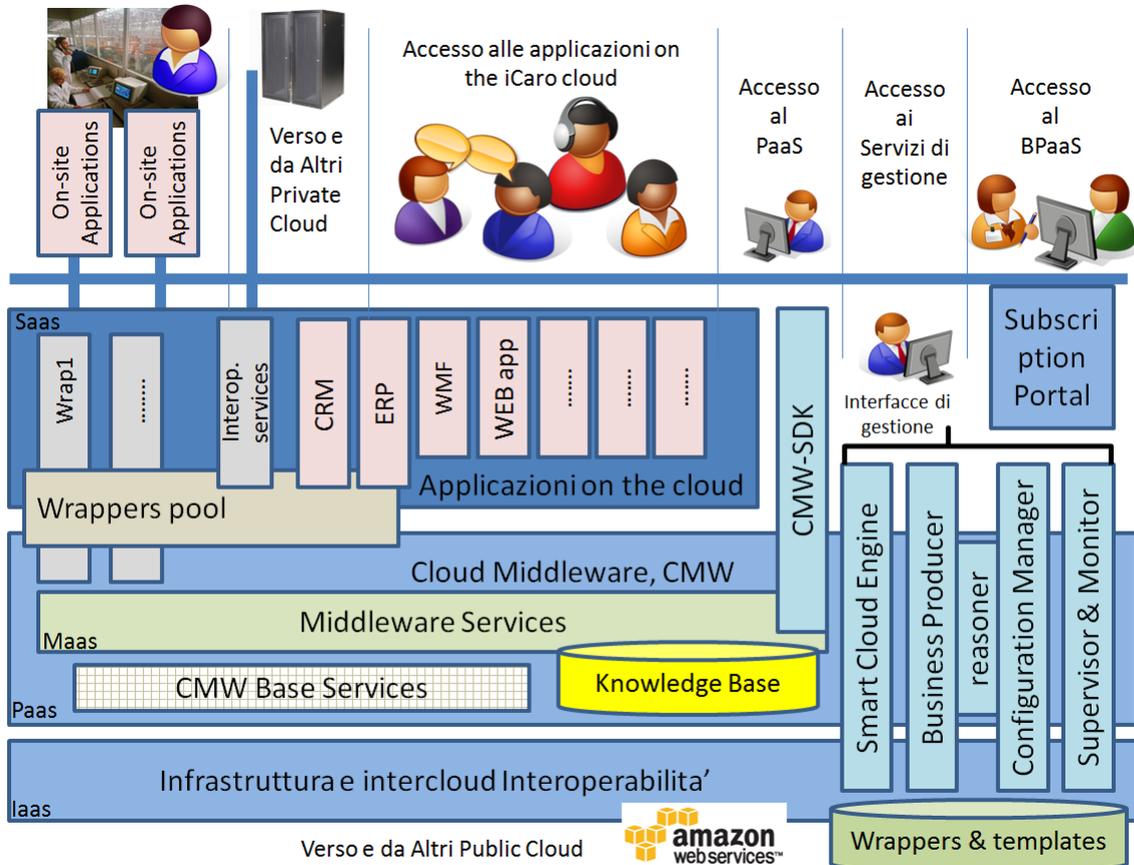


Figura 1 – Architettura di iCaro

Nell'architettura sono identificati i livelli principali, secondo un'interpretazione comunemente identificata dalla comunità scientifica:

- **Business Process as a Service (BPaaS)¹:** definisce l'erogazione di servizi non esclusivamente riferiti ad ambiti applicativi ma direttamente alle funzionalità di business o di processo, potenzialmente trasversali rispetto alle piattaforme applicative. In pratica consiste in un processo di business mappato interamente nel cloud (composto da servizi, applicazioni web, applicazioni legacy, servizi di integrazione, etc.). È su questo tipo di servizi integrati che **iCaro** punta il suo business. Il processo di business è un pattern di servizi ed include problemi di sicurezza, costi, scalabilità connessione fra local e cloud bidirezionale, il cloud può essere un burst per l'azienda, e può sgravare i costi nel momento del bisogno.
- **Software as a Service (SaaS):** consiste nell'erogazione di servizi applicativi di qualunque tipo, accessibili indipendentemente dalla collocazione e dal tipo di device utilizzato. Non è eseguita un'applicazione proprietaria del cliente, ma il cliente stesso paga il diritto (mediante licenza o canone di affitto) di utilizzo di un'applicazione messa a disposizione dal provider, senza preoccuparsi di come essa venga realizzata e gestita nel cloud. L'unica preoccupazione del cliente in questo caso, oltre ovviamente alla scelta della corretta applicazione che soddisfi le sue

¹ Diversamente dalla classificazione NIST su tre livelli (IaaS, PaaS e SaaS), Forrester Research (2009) considera anche questo quarto livello di servizio che, può estendersi fino ad arrivare all'Everything as a Service (XaaS).



necessità, è quella di gestire il numero di licenze richieste in funzione del numero di utenti. Salesforce.com Customer Relationship Management (CRM) è un esempio di soluzione in cui il software è venduto in modalità as a service.

- **Platform as a Service (PaaS):** rappresenta l'erogazione di servizi applicativi di base come sistemi operativi, middleware, linguaggi, tecnologie di base dati e l'ambiente runtime necessari per eseguire l'applicazione, che quindi rimane l'unica cosa sotto la responsabilità dell'utente, oltre alla definizione del modello (e.g., numero e dimensione dei server, datacenter, caratteristiche del networking) da utilizzare per l'esecuzione dell'applicazione. Google AppEngine è un esempio di Platform as a Service. A livello PaaS viene anche collocato l'insieme dei servizi MaaS, Middleware as a Service.
- **Infrastructure as a Service (IaaS) :** è l'erogazione di servizi infrastrutturali relativi a capacità elaborativa, storage, rete e altri elementi di base assolutamente indipendenti da servizi applicativi di qualunque tipo. Si utilizza quindi l'infrastruttura messa a disposizione dal provider per eseguire la propria applicazione, a fronte di un pagamento in base al consumo dell'infrastruttura stessa, lasciando sotto la responsabilità dell'utente la gestione del sistema operativo, dell'eventuale middleware e della parte di runtime, oltre che dell'applicazione stessa. Amazon EC2 è un esempio di servizio IaaS.

L'architettura iCaro presenta i seguenti componenti principali partendo dalla parte sinistra in alto dove i servizi cloud BPaaS sono acquisiti da parte del potenziale utente finale.

- **Subscription/Sales Portal (SP).**
- **Supervisor & Monitor (SM)**
- **Configuration Manager (CM),**
- **Business Producer (BP):**
- **Smart Cloud Engine (SCE)**
- **Knowledge Base (KB)**
- **Cloud Middleware, CMW.**

Infrastruttura hardware necessaria per iCaro include il supporto per circa 50 VM in varie versioni. In molti esperimenti sarà necessario attivare due sezioni del datacenter, svariati cluster, una per la validazione ed una per lo sviluppo. Queste dovranno essere in rete fra di loro per permettere il cloning/moving e l'aggiornamento fra storage distinti:

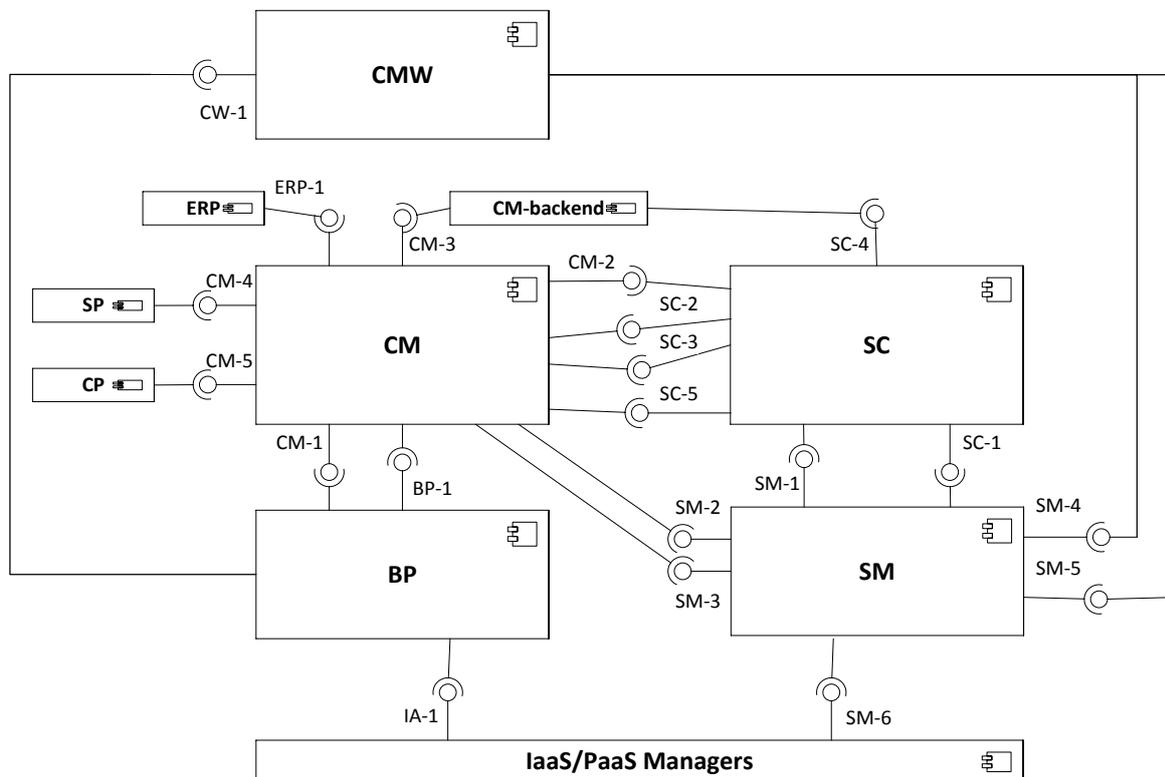
- **cluster e server con caratteristiche di high availability: ridondanza, move, etc.**
- **10 server blade, alimentazione ridondata, 4/8 schede di rete 10 Gbps, 2 CPU (6 core), 24-48 Gbyte Ram**
- **Storage: 50-60 Terabyte, Raid 50/60, partizionato in due storage distinti**
- **Sistema con doppio UPS multiplo con schede di rete**
- **Switch di rete multipli per partizionare la rete**
- **Firewall multipli**
- **Armadio ed accessori**
- **Software di sistema come VMware, IBM come descritto nell'Attività 3.4.**
- **Connessione ad alta velocità**



2. Architettura Generale

La presente sezione riporta l'architettura UML (*component diagram*) di Icaro, e la descrizione delle funzionalità esplicate dai singoli componenti.

Per la modellazione generale sono stati presi in considerazione i macroblocchi costituenti l'infrastruttura Cloud (SP, CMW, BP, SC, SM), e le interazioni fra di essi. Per ogni connessione fra i moduli principali sono stati specificati i protocolli di comunicazione utilizzati o il tipo di chiamate. Di ciascun modulo principale è stata specificata la struttura interna insieme alle funzionalità di ciascun sottomodulo costituente. L'architettura generale di Icaro è rappresentata dal seguente diagramma UML delle componenti.



Di seguito si riporta una legenda dei simboli utilizzati nei diagrammi riportati.

Connessione	Client	Server	API	Descrizione	Protocolli	Doc ID
CM-1	BP	CM	CM-API	Notifica al CM che un servizio è stato istanziato e gli passa i parametri aggiornati.	API REST	3.5, 3.11.1
CM-2	SC	CM	CM-API	SCE Comunica al CM le azioni da intraprendere per riconfigurare i servizi e deploy.	API REST	3.4, 3.12.1, 3.5
CM-3	CM-backend	CM	CM-API	Notifica al CM la presenza di un nuovo strumento/servizio.	API REST	3.5, 3.12.1, 3.2
CM-4	SP	CM	CM-	Inserisce/Aggiorna il CM	API	3.5, 3.12.1



			API	con i dati di SP.	REST	
CM-5	CP	CM	CM-API	Inserisce/Aggiorna il CM con i dati di CP.	API REST	3.5, 3.12.1
ER-1	CM	ERP	ERP-API	Comunica all'ERP i dati di fatturazione.	API REST	3.5, 3.12.1
BP-1	CM	BP	BP-API	Chiede al BP di fare il deploy dei servizi.	API REST	3.5, 3.11.1, 3.12.1
SC-1	SM	SC	SC-API	SM salva i dati di monitoraggio su KB.	API REST	3.4
SC-2	CM	SC	SC-API	CM registra/aggiorna le configurazioni con parametri aggiornati.	API REST	3.4
SC-3	CM	SC	SC-API	CM controlla le configurazioni.	API REST	3.4, 3.12.1, 3.5
SC-4	CM	SC	SC-API	Interfaccia per l'inserimento/aggiornamento della definizione di un'applicazione	API REST	3.4, 3.12.1, 3.5
SC-5	CM	SC	SC-API	Interfaccia per l'inserimento/aggiornamento della definizione di un datacenter.	API REST	3.4, 3.12.1, 3.5
SM-1	SC	SM	SM-API	Inserisce/Aggiorna le configurazioni di monitoraggio sull'SM.	API REST	3.4
SM-2	CM	SM	SM-API	Chiede dati di monitoraggio all'SM.	API REST	3.4, 3.12.1, 3.5
SM-3	CM	SM	SM-API	Chiede grafici di monitoraggio all'SM.	API REST	3.4, 3.12.1, 3.5
SM-4	Collectors	SM	SM-API	Comunica all'SM i dati raccolti.	API REST	3.4
SM-5	Agents	SM	SM-API	Comunica all'SM i dati raccolti.	API REST	3.4
SM-6	IaaS Managers	SM	SM-API	Comunica all'SM i dati raccolti.	API REST	3.4
CW-1	BP	CMW	CMW-API	Inserisce/Aggiorna Servizio.	API REST	3.5, 3.11.1, 3.2
IA-1	BP	IaaS/PaaS Manager	several	Deploy/Aggiornamento delle risorse.	several	3.5, 3.11.1

L'architettura generale di Icaro è composta dai seguenti componenti principali:

- **Subscription/Sales Portal (SP)**, portale dei servizi di vendita **Icaro**, sottoscrizione e monitoraggio del servizio acquistato per i parametri di acquisto, simulazione di acquisto, firma del contratto, etc.
- **Business Producer (BP)**, tool per la configurazione dei pattern/configurazioni di deploy complesse composte da: VM, applicazioni, servizi sulla base di best practice, profiling, etc.
- **Supervisor & Monitor (SM)**, sottosistema di supervisione e monitoraggio dei servizi sul cloud, sia per gli amministratori che per i singoli clienti tramite un interfaccia che esporta i dati verso il SP-fe.
- **Smart Cloud (SC)**, motore autonomo per la supervisione, l'automazione e l'ottimizzazione dei servizi e dello sfruttamento del cloud.



- **Cloud Middleware (CMW)**, implementa una soluzione PaaS di tipo evoluto rispetto alle soluzioni di mercato allo stato dell'arte, che tipicamente sono orientate a tipologie limitate di applicazioni e servizi di base, come per esempio ai soli servizi di registrazione e alle applicazioni web su cloud
- **Cloud di riferimento**, infrastruttura cloud di iCaro composta da svariati host e macchine virtuali, template, etc. etc. A livello di IaaS vi sono sia VMware/vCenter che VMM di MS.
- **SandBox**, una porzione di infrastruttura che replica un piccolo cloud per il test e la validazione delle applicazioni e dei servizi in preparazione.

La specifica dell'interfaccia e delle API rest è a carico del responsabile del singolo elemento che la contiene. Pertanto, la specifica può essere reperita nel documento corrispondente.

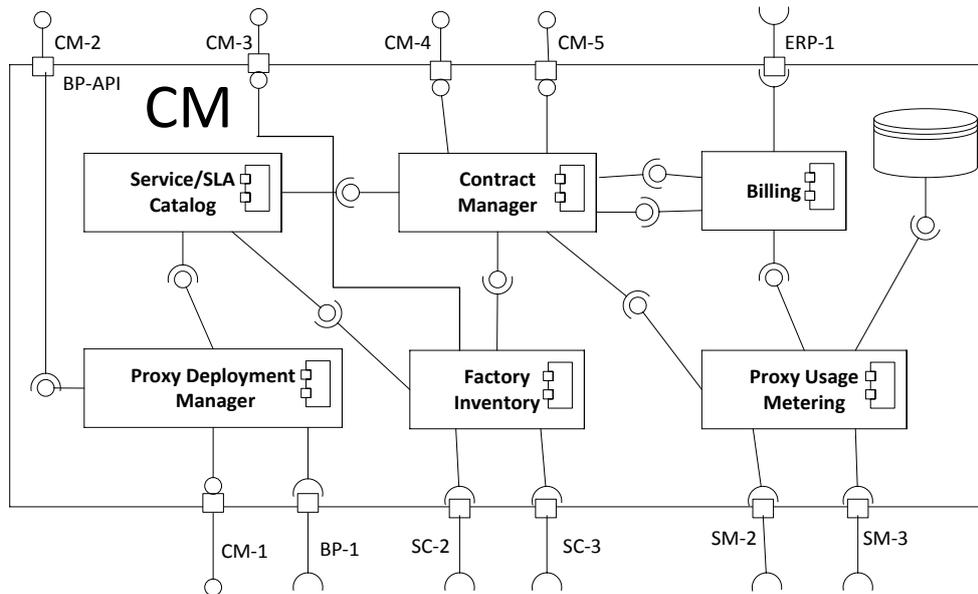
In dettaglio l'architettura generale di ICARO è composta dai componenti specificati nelle seguenti sezioni.

2.1 Subscription Portal (SP)

- **Subscription/Sales Portal (SP)**, portale dei servizi di vendita **iCaro**, sottoscrizione e monitoraggio del servizio acquistato per i parametri di acquisto, cambio della configurazione del servizio, definizione della **SLA** (service level agreement), simulazione di acquisto, firma del contratto, etc. Il SP permette di accedere al catalogo dei servizi e di comporre la propria configurazione e scelta in modo guidato ed intelligente. Il SP permette in maniera automatica/assistita la tariffazione e la personalizzazione dei servizi a corredo alla soluzione desiderata (questo implica aver predisposto un offerta di servizi cloud e di prodotti su cloud). Sulla base della configurazione richiesta (fra quelle possibili e proposte) sono predisposti i modelli di costo e le possibili SLA, tenendo conto delle innumerevoli variabili di costo, configurazione, stato del cloud, etc. SP si compone di alcune sottoparti:
 - **SP-fe (front end)**, front end di vendita, con registrazione degli utenti, carrello, pagamento e-commerce, etc.
 - presenta una sorta di CRM in cui il cliente può vedere lo stato dei suoi servizi attivi, gli andamenti dei parametri inclusi nel servizio (per esempio, numero di utenti, storage, carico e traffico sulla rete, etc.), etc. Per acquisire tali andamenti effettua una chiamata ai servizi dell'SM che fornisce un segmento HTML con tutti i grafici relativi ad una certa SLA per un certo utente X.
 - SP-fe, potrebbe includere la creazione in modalità self-service di una soluzione in maniera grafica drag-n-drop.
 - **SP-admin**, gestione amministrativa delle vendite, degli utenti, etc.
 - per comprendere la disponibilità di vendita e pertanto le SLA formalizzate per i cari prodotti in vendita accede alla Knowledge Base, emette una richiesta al Reasoner o allo SCE per capire lo spazio disponibile per le varie configurazioni messe a catalogo (per esempio dammi la disponibilità per Sigla ++, sulla base di Host, licenze, etc.).
 - tali verifiche possono anche essere effettuate off-line, per esempio una volta al giorno.
 - tali informazioni possono anche essere memorizzate localmente in un DB di vendita ma le informazioni attuali sono presenti nella KB.



2.2 Configuration Manager (CM)



La figura precedente riporta il diagramma UML del componente Configuration Manager (CM), costituito dai seguenti componenti:

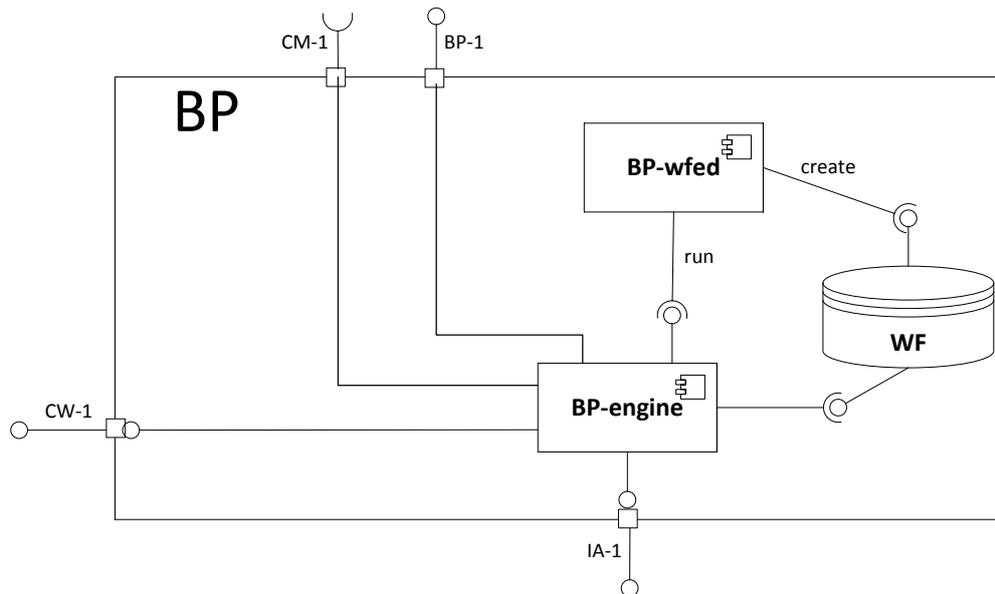
- **CM/SLA-Editor**, strumento amministrativo per la definizione e formalizzazione delle SLA in qualche formato formale XML. **Configuration Manager (CM)**, tool per la valutazione delle possibili configurazioni del sistema, effettua la verifica della consistenza e delle completezza tramite modelli formali e effettua la generazione della logica/flusso di produzione / deploy della configurazione scelta (questa a sua volta viene eseguita dal Business Producer). Effettua il cambio della configurazione del servizio, e la definizione della **SLA** (service level agreement).
 - tale editor deve salvare le informazioni delle SLA prodotte ed associate alle configurazioni nella KB. Le SLA formalizzate vengono salvate in locale allo SP e passate al BP come file che a sua volta le passa alla KB quando viene completato un Deploy con successo. La SLA modello ha senso anche su SP mentre la SLA effettiva associata ad un certo Workflow, con nome del cliente, etc. ha senso solo se gestita tramite il BP e passata alla KB quando il deploy è andato in porto.
 - per la produzione delle SLA l'utente esperto deve conoscere il modello di deploy ed il modello di business. Pertanto si prevede che la definizione della SLA ed il suo editing siano effettuate in stretto contatto con la definizione del flusso di deploy del BP.
 - potrebbe essere necessario avere dal BP una lista di parametri di costo per ogni Workflow di deploy. Per esempio per ogni configurazione possibile: lista macchine virtuali con le loro caratteristiche, lista servizi accesi, carico su disco, carico di rete, accesso a servizi esterni alla VM, etc. In modo da essere sicuri che nella definizione della SLA non si vada a perdere di vista qualche aspetto di costo importante.
 - il **CM** sfrutta un **Reasoner** (SC) che utilizza soluzioni di intelligenza artificiale e semantic computing: effettua verifiche e validazioni (V&V). Reasoner viene attivato quando vengono fatte delle nuove configurazioni/pattern da parte del BP e/o SLA da parte dello **SLA-Editor**.
 - **Verifica**, un confronto fra la SLA associata ad una configurazione e la descrizione della configurazione stessa potrebbe essere svolta dal Reasoner (SC), in modo da mettere in evidenza eventuali dimenticanze e/o inconsistenze. Questa verifica la può fare il Reasoner quando il BP invia la configurazione e la SLA di una nuova attivazione. Il Reasoner dovrebbe anche verificare che i parametri della SLA siano in effetti monitorati con successo.



Vi è inoltre un ulteriore componente, chiamato **CM-backend**, che notifica al CM le configurazioni e le SLA.

- CM-Backend fornisce la configurazione di una soluzione attraverso un'interfaccia fornita da KB (SC). Al momento il KB si aspetta in ingresso un XML contenente una OVL con la descrizione in termini di triple.

2.3 Business Producer (BP)



La figura precedente riporta il diagramma UML del componente Business Producer (BP), costituito dai seguenti componenti:

Business Producer (BP), tool per la configurazione dei pattern/configurazioni di deploy complesse composte da: VM, applicazioni, servizi sulla base di best practice, profiling, etc. Il BP è un'evoluzione dei tradizionali BPM per cloud, è in buona sostanza un orchestrator evoluto, che tramite descrizioni di flusso permette di formalizzare la procedura di deploy delle applicazioni e dei servizi sul Cloud. In **iCaro**, il modello viene esteso a lavorare con sistemi complessi composti da: servizi di base, applicazioni WEB, applicazioni legacy e servizi per applicazioni on site remote per cloud bursting e bridging. Si compone di alcune sottoparti:

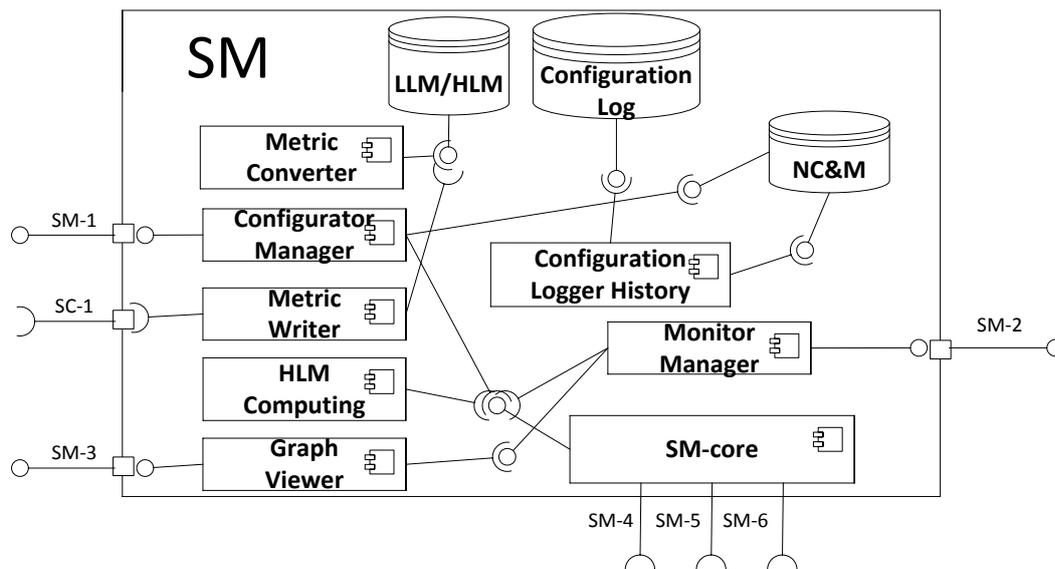
- **BP-wfed (workflow front end)**, BP workflow editor per la formalizzazione e collaudo dei workflow di deploy di una configurazione complessa, un business. Questo componente crea workflow che vengono messi in esecuzione da un BP-Engine.
- **BP-Engine**, esegue i workflow procedendo al deploy vero e proprio. Durante il flusso di deploy si devono tenere conto di alcune operazioni da svolgere:
 - **verifica** delle presenza di risorse per procedere alla produzione descritta nel flusso di deploy, workflow. Verifica che le risorse necessarie siano presenti e disponibili.
 - **istanzia** VM, e applicativi, servizi, etc. etc. Prima verifica, poi istanzia, in seguito registra su SM ed infine aggiorna la KB. Se vi sono problemi deve poter tornare indietro rimuovendo le tracce dei vari passi.
 - per istanziare fa uso dei template, wrapper e intaller messi a disposizione nel pool dei componenti possibili. Tale Pool deve essere conforme al BP in uso. Sarà cura del CMW fornire tali informazioni per il Pool.
 - per istanziare a livello IaaS ovviamente utilizza il gestore della parte IaaS, per esempio Vcenter, VMM di MS. Chiede a livello IaaS dove poter mettere tali VM, etc. etc. per esempio utilizzando un Resource Pool.



- **inviare la configurazione** alla KB in modo da registrare il vero deploy, sia come tipologia che come istanza specifica, con i relativi ID di SLA, la SLA stessa, i parametri, gli iD del cliente, etc. etc.
- **registrare** sulla KB quale sia la SLA di riferimento della configurazione specifica, oppure se questa è già associata a tale configurazione citare tale configurazione in modo univoco.
- **registrare** i servizi, le macchine, etc. etc. che devono essere monitorate e con che parametri, tale registrazione viene svolta con opportune chiamate verso SM
- **avere dei servizi aggiuntivi** chiamando il CMW-Service-Directory. A fronte di nuovi servizi o di perdita di connessione), il CMW-SD rilancia istanze di CMW-Collector per garantire che tali servizi e Software Applicativo con metriche di alto livello siano monitorate in modo opportuno.
- **verificare** la consistenza della configurazione, effettua un ciclo di test, per esempio fa delle chiamate a WS, chiamate REST, si aspetta della risposte e dei valori cambiati, li verifica se sono tali.
 - in caso negativo tutto l'installato deve essere rimosso oppure viene chiamata assistenza, mettendo in evidenza la configurazione allo stato corrente, il problema riscontrato, eventualmente come fare per rimuovere il tutto (procedura di disinstallazione).
- **Attivazione del BP-Engine.** Il BP-Engine è lo strumento che mette effettivamente in esecuzione un Workflow di deploy, lo interpreta. Questo può essere attivato dall'esterno ed in particolare da:
 - **SCE** qualora vi sia
 - per un certo client/SLA, in base alla SLA specifica (il contratto con il cliente) la possibilità di cambiare dinamicamente configurazione. Per esempio più banda, più disco, configurazione diversa.
 - L'occorrenza di condizioni critiche ripetute, per esempio una sfioramento dei limiti per N volte, un carico eccessivo di rete già notificato più volte al cliente, etc. etc.
 - **SP** qualora un cliente verifica la configurazione di un servizio che ha acquisito e la voglia cambiare, oppure tramite SP si sia venduta una nuova configurazione e/o un nuovo prodotto. Questo può implicare la verifica che vi sia effettivamente risorse per la produzione o meno e la richiesta di una produzione verso il BP.
- **BP-Ingestor**, questo modulo del BP effettua la registrazione di eventuali template, installer, wrapper, etc., per processi classici su cloud oppure per bursting, bridging, etc. Questo modulo permette di mettere a disposizione per il BP-wfed (Workflow Editor) nuovi template, applicazioni, servizi, virtual machine, etc. etc. Tipicamente un semplice ingestor a servizio dell'orchestrator. Tipicamente Solo i servizi registrati e descritti in modo proprio possono essere composti ed utilizzati nella produzione in workflow.

2.4 Supervisor and Monitor (SM)

La figura seguente riporta il diagramma UML del componente Supervisor and Monitor (SM).



Supervisor & Monitor (SM) sottosistema di supervisione e monitoraggio dei servizi sul cloud, sia per gli amministratori che per i singoli clienti tramite un'interfaccia che esporta i dati verso il SP-fe. Include la connessione con il sistema di controllo standard a livello IaaS tramite strumenti di mercato come per esempio la suite di soluzioni VMware e/o VMM di MS che permettono di monitorare le risorse del cloud a livello host e VM: rete, cpu, memoria, storage, configurazione, etc.;

Si compone di alcune sottoparti:

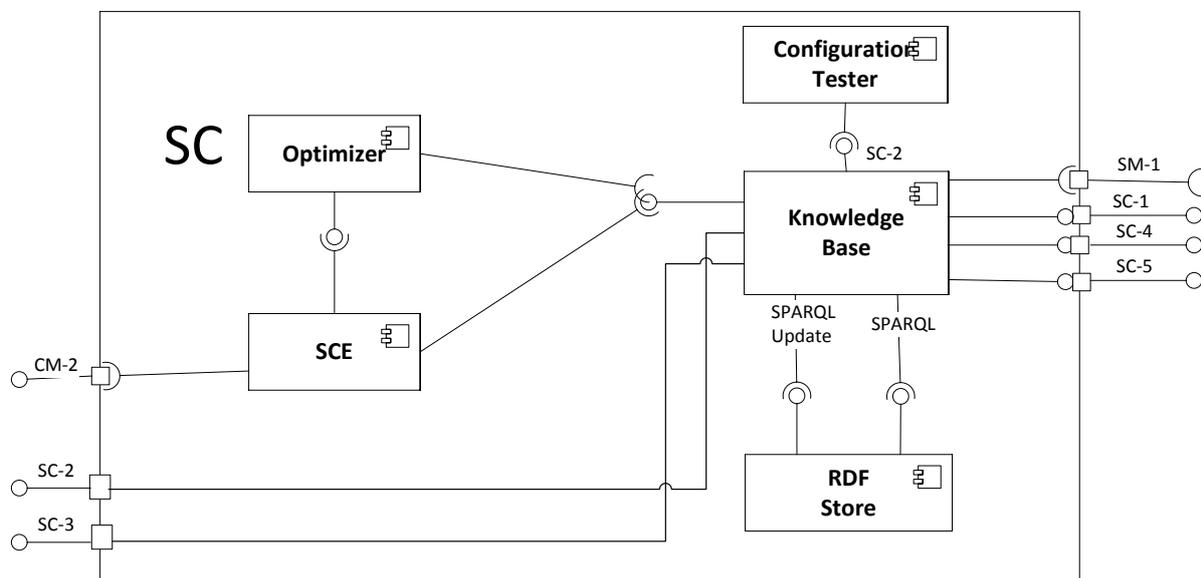
- **SM Configurator Manager**, interfaccia tramite la quale SM riceve richieste (con relative informazioni formalizzate) per aggiungere il monitoraggio di VM, servizi, etc. Tali richieste di monitoraggio vengono inviate da parte del BP durante la produzione di un deploy e/o di un cambio di configurazione, e sono relative ad un certo cliente, per una certa SLA, etc. etc..
- **SM-core**, strumento di monitoraggio basato su Nagios.
 - permette la configurazione del monitoraggio delle risorse a livello IaaS. L'arrivo di un nuovo Host e di nuove schede e connessioni di rete, la presenza di un nuovo Storage vengono registrati tramite l'interfaccia di SM-Core.
 - richiede e colleziona dati di monitoraggio registrati tramite richieste che provengono dal BP via **SM Configurator Manager** e relative al monitoraggio di VM, servizi di base, servizi acceduti tramite CMW Collector vari, etc. Tutti questi dati vengono collezionati tramite SM-Engine.
 - questo engine è di fatto realizzato con uno o più processi Nagios di monitoraggio classico. Sono processi che si vanno a leggere i dati verso Software, Servizi in rete, ODBC, VM, gestori di VM, etc. Tutto tramite protocolli standard presenti in Nagios. Sono inclusi chiamate ai servizi del CMW che vengono letti accedendo al CMW Collector.
- **HLM Computing**, processo di stima automatica delle metriche di alto livello sulla base delle metriche di basso livello collezionate in modo automatico da agenti e agentless nel SM.
 - Sono metriche di alto livello quelle che non sono rilevate direttamente ma devono essere computate.
 - Per esempio: la media del carico di rete, la somma della transazioni fatte negli ultimi 14 mesi, il numero di sforamenti della SLA nell'ultimo mese, etc. etc.
- **Metric Writer**, strumento di esportazione periodica dei dati importanti dal mondo SM verso la KB, da SQL a RDF, attraverso il **Metric Converter**) per eventuali ragionamenti e deduzioni da svolgere su KB da parte del Reasoner. Fra questi dati:
 - allarmi e sforamenti
 - evoluzioni e stato della configurazione IaaS
 - metriche di alto livello associate a servizi e a SLA



- **Graph Viewer**, interfaccia di export dei grafici relativi ai parametri citati nelle SLA corrispondenti per un certo Cliente/SLA, export che viene tipicamente effettuato verso SP. Questo export include grafici relativi a tutti i parametri, per esempio, la rete, il numero di utenti, etc. e viene prodotto su base mensile, settimanale, attuale, etc. L'ultima settimana, l'ultimo mese, l'ultimo anno, etc.
- **SM Monitor Manager**, interfaccia tramite la quale SM riceve richieste (con relative informazioni formalizzate) per l'acquisizione di valori puntuali di metriche.
- **Configuration Logger History**, traccia tutte le chiamate in arrivo sull'SM Configurator Manager (scrivendole in una database), realizzando un history log per ogni configurazione presente nel sistema di monitoraggio.

2.5 Smart Cloud (SC)

La figura seguente riporta il diagramma UML del componente Smart Cloud (SC), costituito dai seguenti componenti:



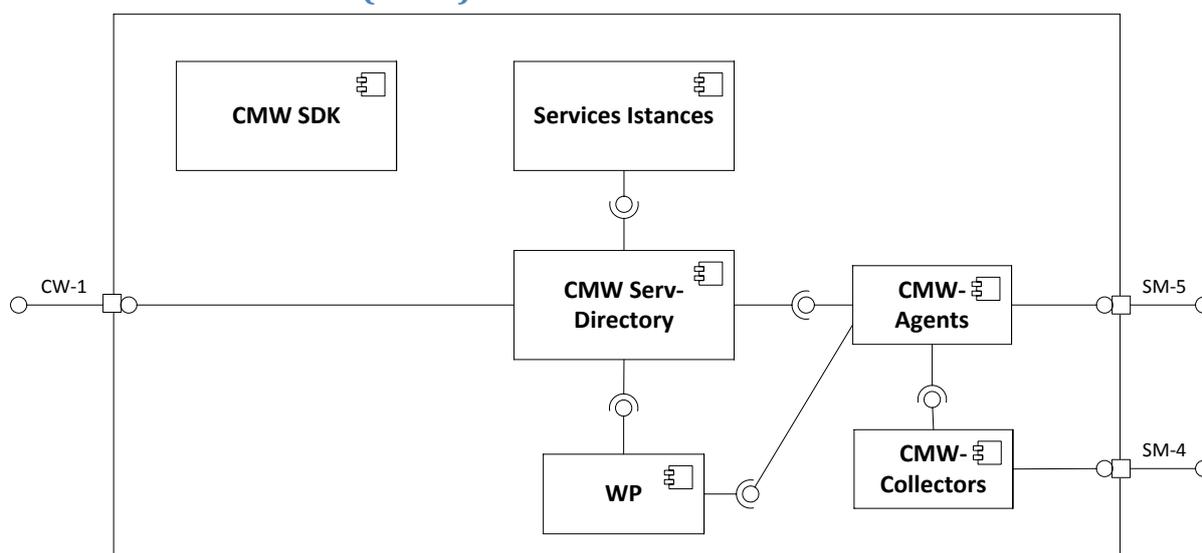
Lo Smart Cloud di Icaro è costituito dai seguenti componenti:

- **Smart Cloud Engine (SCE)**, motore autonomo per la supervisione, l'automazione e l'ottimizzazione dei servizi e dello sfruttamento del cloud. Lo **Smart Cloud Engine**:
 - **formalizza funzioni di controllo sui dati di** monitoraggio dei livelli IaaS, PaaS, SaaS, e dal CMW, che tramite SM finiscono dentro la KB. Si basa su una logica ad eventi e su condizioni di Firing che permettono di arrivare a prendere delle decisioni. L'insieme delle condizioni deve essere dimostrato consistente. Si prevede l'utilizzo di soluzioni di semantic computing, modelli formali. Questa astrazione rappresenta il centro dell'intelligenza del progetto ICARO e consente di slegare l'infrastruttura fisica sottostante dalle soluzioni on-top.
 - **interpella il Reasoner** per l'esecuzione **periodica e/o sporadica** di regole ed inferenze sulla KB di ICARO. Regole e query che **realizzano politiche ed automatismi per**
 - **la gestione di** emergenze e/o
 - inneschi e/o
 - verifiche di configurazione anche sulla base delle risorse disponibili sul CLOUD.
 - **associa** alle query/regole eventuali azioni che possono scaturire in base ai risultati e alle deduzioni prodotte dal Reasoner e/o dall'Ottimizzatore:
 - attivazioni di configurazioni e riconfigurazioni tramite il BP
 - notifiche e/o allarmi verso i/il Clienti/e e/o l'Amministratore(i)



- attivazioni di altre regole/query sul Reasoner
 - attivazioni di processi di Ottimizzazione per la stima di altre configurazioni possibili con parametri diversi, e/o per la verifica di condizioni e di efficienza. Interpella un eventuale **Ottimizzatore** qualora vi siano i presupposti per poter ottimizzare il carico del cloud in qualche misura e/o per effettuare controlli di efficienza
 - alcune delle deduzioni delle regole possono portare a decidere di usufruire di risorse di altri datacenter cloud privati o pubblici (es. per effettuare una migrazione dei dati in maniera trasparente all'utente finale, oppure per usufruire di ulteriori capacità di calcolo da altri cloud provider, con integrazioni di cloud pubblici quali Amazon, Azure, ma anche privati aziendali);
- **Knowledge Base (KB)**, si fonda sull'ontologia descrittiva del cloud **iCaro**, contiene e gestisce in un database semantico. La KB viene interrogata da un **Reasoner**. La KB contiene:
 - la descrizione dell'infrastruttura in termini di host, cpu, rete, etc. Questi dati li riceve da SM.
 - catalogo dei servizi e applicazioni di **iCaro** come i servizi di base, applicazioni WEB, applicazioni legacy on cloud, servizi di cloud bursting per applicazioni on site legacy, servizi di interoperabilità. I servizi devono essere descritti e formalizzati tramite descrittori formalizzati con una ontologia in modo da poter usare queste informazioni per fare delle deduzioni tramite il Reasoner;
 - i descrittori delle applicazioni come configurazioni, template, scalability, pattern, etc. regole di produzione e di scaling
 - stato del cloud in termini di business allocati, cluster da controllare, regole di scaling, etc.
 - dati di consumo e monitoraggio (database relazionale classico tracciamento delle misure, dello stato, delle variazioni, etc.);
 - definizione dei modelli di business, SLA/Dynamic SLA.
 - **Reasoner**, esegue query e verifiche sulla base dell'insieme di queste verifiche e regole che sono state definite dallo SCE. Queste query e verifiche possono essere periodiche o aperiodiche.

2.6 Cloud Middleware (CMW)



La figura precedente riporta il diagramma UML del componente Cloud Middleware (CMW), costituito dai seguenti componenti:

Cloud Middleware (CMW), questo componente implementa una soluzione PaaS di tipo evoluto rispetto alle soluzioni di mercato allo stato dell'arte, che tipicamente sono orientate a tipologie limitate di applicazioni e servizi di base, come per esempio ai soli servizi di registrazione e alle applicazioni



WEB su cloud. In **iCaro** questo strato permette di astrarre i servizi cloud per: (i) servizi di base specifici come mailing, certificazione, e-commerce, backup, storage, computing on demand; (ii) applicazioni on the cloud; (iii) applicazioni legacy on site; (iv) poter controllare e monitorare accesso e consumo dei servizi; (v) accedere a soluzioni di cloud bursting e bridging. Alcune applicazioni possono dover venire adattate per usare questi servizi, per questa ragione deve essere sviluppato uno specifico tool di sviluppo, **CMW-SDK**. In alcuni casi, l'adattamento potrà essere possibile tramite l'uso di **template** VM, di **wrapper** di servizi, di servizi SOA (WS, rest, etc.), di proxy, etc. Anche queste soluzioni possono essere studiate e predisposte per essere disponibili all'uso nel sistema di configurazione automatizzata del sistema cloud di **iCaro**; Si compone di alcune sottoparti:

- **CMW-SDK**, kit di sviluppo in vari linguaggi per applicazioni e servizi che possono essere deployati su ICARO e monitorati di conseguenza. Il CMW-SDK utilizza una SandBox per lo sviluppo di nuove Applicazioni (Sx in figura) e/o di nuovi servizi e/o di nuove VM o che altro. Queste vengono finalizzate ed esportate in forma di: templates, installer, wrapper, etc. Associati a queste informazioni si devono affiancare descrittori di SLA per poter finalizzare anche la vendita di tali servizi e tool sempre in ottica Cloud. Per completare la SandBox di test, è probabile la necessità di accedere ai servizi del BP per richiedere il deploy di un ambiente o di una singola VM in un'area dedicata del Cloud.
- **CMW-Service-Directory**, tool di registrazione dello stato dei servizi disponibili, attivazione di nuovi servizi, consistenza dei servizi attuali, messa in esecuzione di CMW Collectors in modo da creare una soluzione scalabile composta da collettori di dati ridondati. Il servizio di back up come i wrapper e come il monitoraggio di dati a consumo (numero di fatture, numero di utenti, etc.) sono di fatto servizi e vengono elencati nella Service Directory.
- **CMW-Active**, insieme dei servizi disponibili, tipicamente allocati su più macchine virtuali e su più host. La cui lista è gestita da **CMW-Service-Directory**.
- **CMW-Collectors**, collezionatore di dati da servizi a consumo, specialmente quelli relativi alle applicazioni che sono a livello SaaS che per esempio utilizzando servizi del CMW a consumo o che implementano modelli di business a consumo: sul numero di fatture, sul numero di utenti, etc. Il conteggio di tali informazioni passa sempre tramite un servizio di conteggio creato a livello di **CMW-Active**.