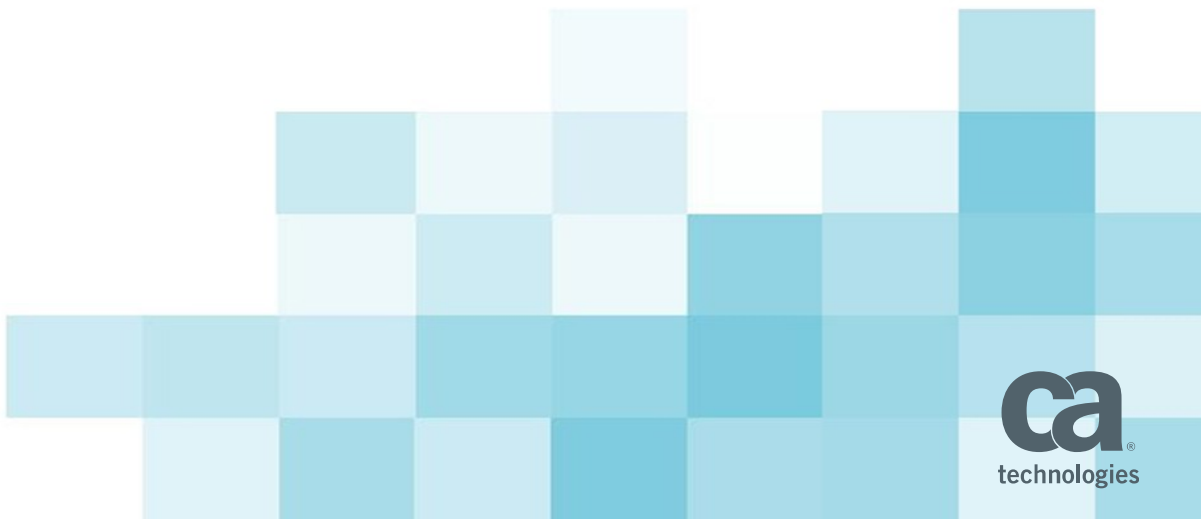


# Les meilleures pratiques d'usage

## CA APM 10

Joerg Martin  
APM Global SWAT

**Janvier 2017**



# Défis actuels

- Manque de processus/planification compliquant le déploiement APM
- Mauvais déploiement entraînant des problèmes majeurs de performance et de stabilité
- Lacunes dans la surveillance résultant en une valeur réduite de APM
- Manque de suivi résultant en une valeur réduite de APM

# Questions à se poser avant le déploiement

## Identifier la qualité de déploiement

- Le système APM est-il dimensionné correctement?
- APM CEM sera-t-il également utilisé?
  - Le système APM/CEM est-il installé au bon endroit?
- A-t-il été décidé du rôle des utilisateurs?
- Qui fera la suivi quotidien de APM?
- Qui recevra les rapports?

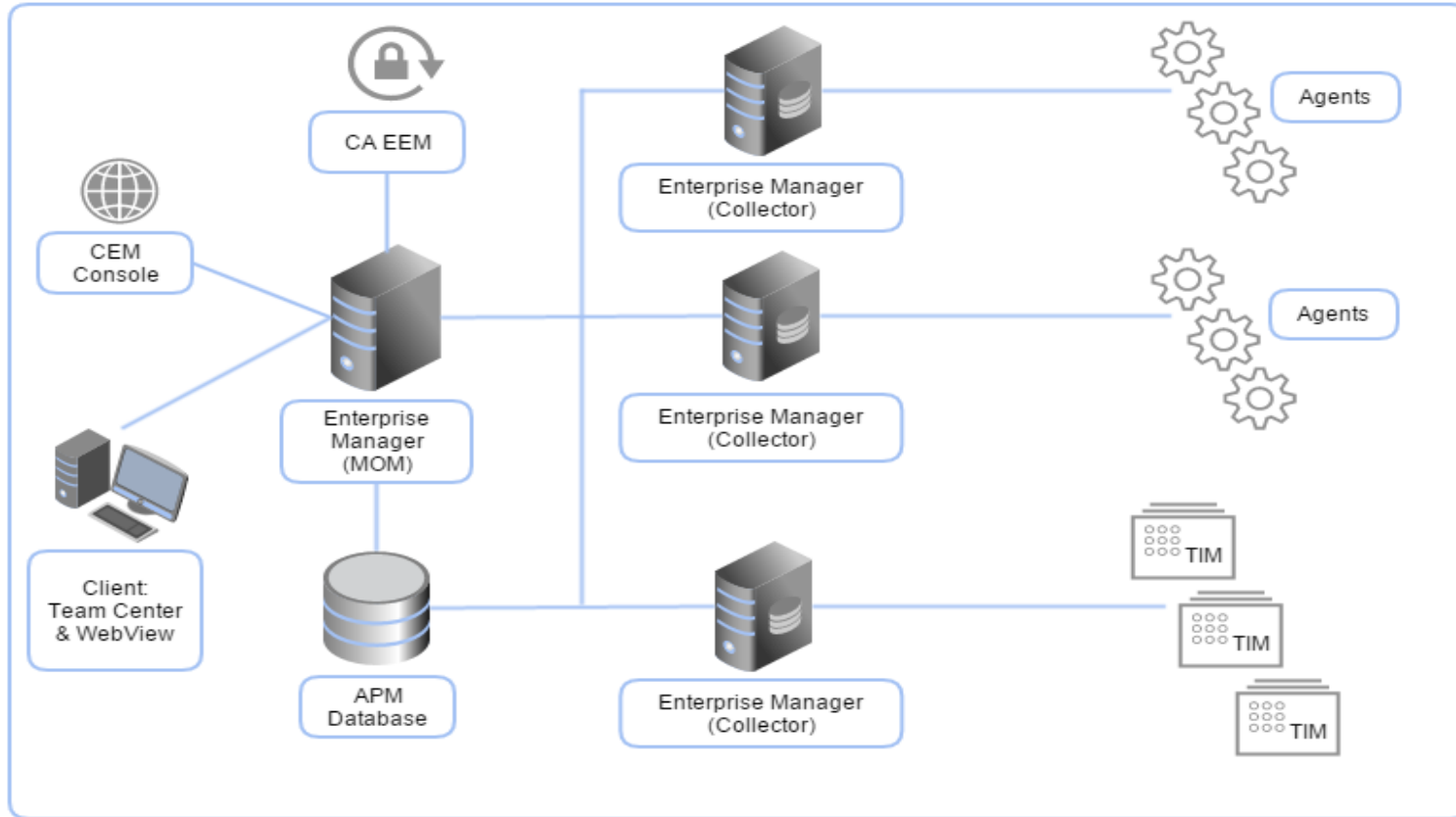
# Vérifier: Dimensionnement de APM

- Une évaluation des métriques attendues/probables a-t-elle été effectuée?
- Déterminer s'il s'agit d'un petit, moyen ou grand déploiement
  - Pensez aussi à la croissance future
- S'assurer que CPU, mémoire, débit E/S disque sont suffisant
- Vérifier les ressources matérielles, la compatibilité, le débit E/S, etc.

# Vérifier: Installation de APM

- Un dessin technique de déploiement a-t-il été fait?
  - Pour voir les flux de communication entre composantes APM
  - Pour placer une sonde CEM
- Les pare-feu ont-ils été configuré pour permettre l'intercommunication des composantes APM?
- Qui est responsable des différentes composantes APM (matériel), Pare-Feu, logiciels, sécurité ...

# Vérifier: Installation de APM



# Vérifier: Répartition des tâches (Rôles)

## Spécialiste des applications

- Pas forcément un spécialiste APM, mais aura beaucoup à interagir avec APM
- Gère le cycle de vie des applications/logiciels pour les sponsors

## Chef de Projet

- Gère l'initiative APM, surtout en ce qui concerne les questions de personnel

## Administrateur APM

- Installe et configure les composantes APM

## Architecte APM

- Stratégie de surveillance, sélection d'outils, conception de processus, normes technologiques, normes d'éducation interne, dimensionnement des solutions

## Specialist APM

- Principal contacte pour toutes question sur APM. Se concentrer sur le triage et lutte contre les incendies, les configurations avancées, la gestion des capacités de l'APM et de la planification

# Déploiement: Liste des verifications

- Fair en sorte que les MoM et Collecteurs sont proche (réduire latence de communication)
- Synchroniser l'horloge des systèmes (MoM, Collecteur, TIM) sur la même source
- Dédier un disque à Smartstore
- Dédier un disque à PostgreSQL (si CEM/TIM est utilisé)
- Dédier des collecteurs aux Agent .Net/Batch, aussi à CEM/TIM si utilisé
- Prévoir un collecteur de “rechange” pour les pointes de charge ou dans le cas d'un collecteur défectueux
- Déployer les systèmes un à un, et vérifier les Indicateur clé de performance!



# Déploiement: Liste des verifications II

- Active la API Rest
- Installer les applications similaires cote-à-cote
- Déployer JMAP (partie de JDK) pour des Heap Dump et JStack pour des Thread Dump
- En cas d'installation sur VM/ESX, faire en sorte que les ressources sont dédiées (Mémoire, CPU, E/S disque, espace disque)

# Déploiement: Liste des vérifications III - CEM

- Vérifier la qualité du trafic réseau envoyé à la sonde TIM
- Configurer des transactions à surveiller (éviter des configuration qui se chevauchent)
- Vérifier que le système de la sonde TIM n'est pas surchargé
- Vérifier que l'agrégation de donnée fonctionne (horaire, quotidienne, mensuelle)

# Déploiement: Accès

- Contrôle d'accès
  - Créer Utilisateurs/Roles
    - LDAP, SAML, Local
    - Rôle: Admin, Afficher uniquement, Utilisateur du Tableau de bord
  - Domains.xml, Universes
- Créer des modules de gestion
  - Séparer ces modules en fonction de l'application
- Définir le regroupement des métriques
  - “tuner” pour ce qui n'est pas disponible par défaut
  - Éviter les expressions \*.\*

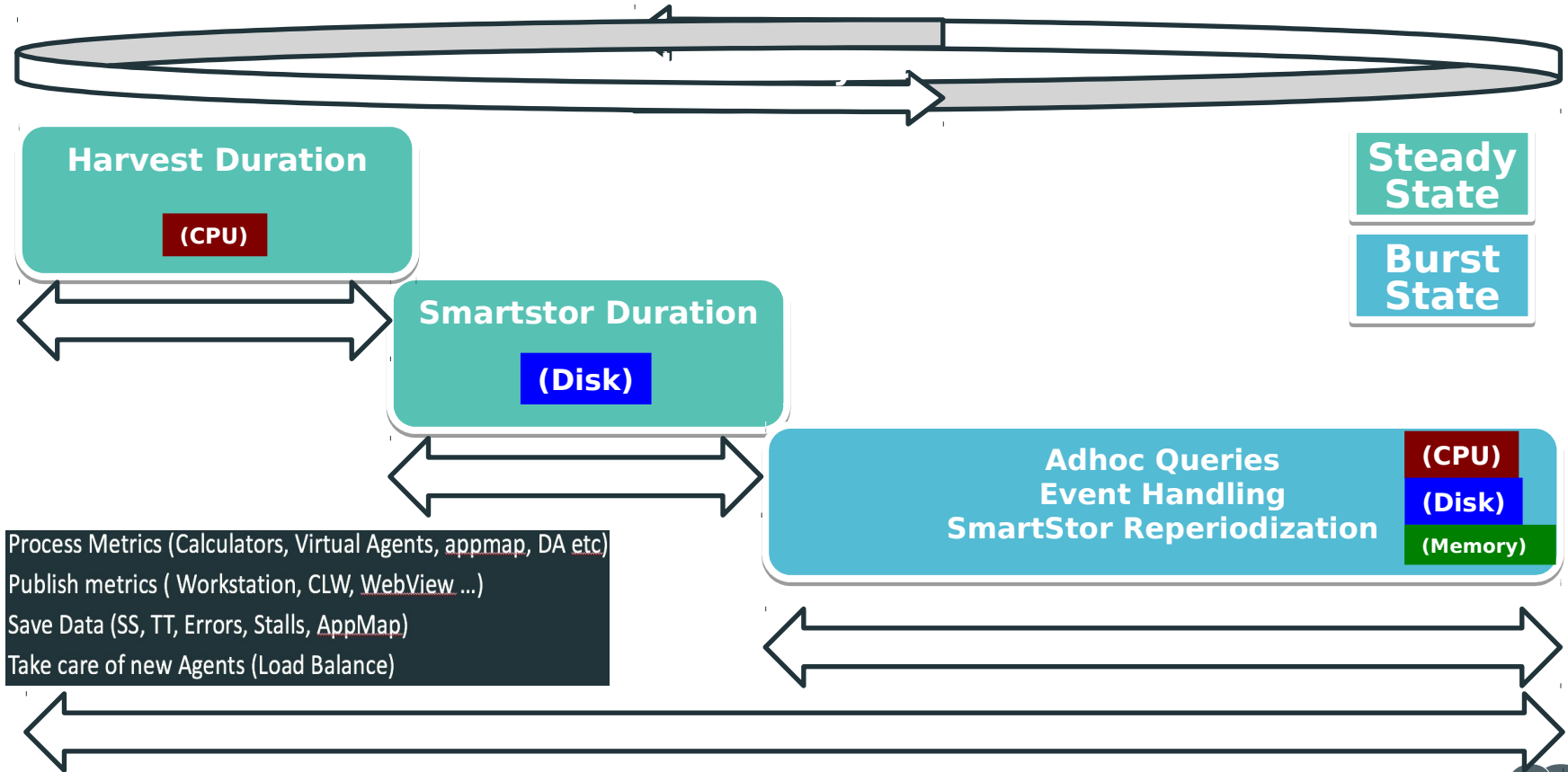
# Déploiement: Actions, Alertes, Dashboards

- Définir les actions
  - Console, SMTP etc.
  - Ajouter des references aux E-Mail modèle (corp, référence URL etc.)
- Définir les alertes
  - Sont définies par default, mais peuvent être personnalisées (ex. OO Memory)
  - Inclure des actions dans les alertes
- Créer des Dashboard pour les différents utilisateurs (Exécutif, niveau composants etc. )
- Définir des calculateur Java si besoin (Eviter les expressions \*.\* )

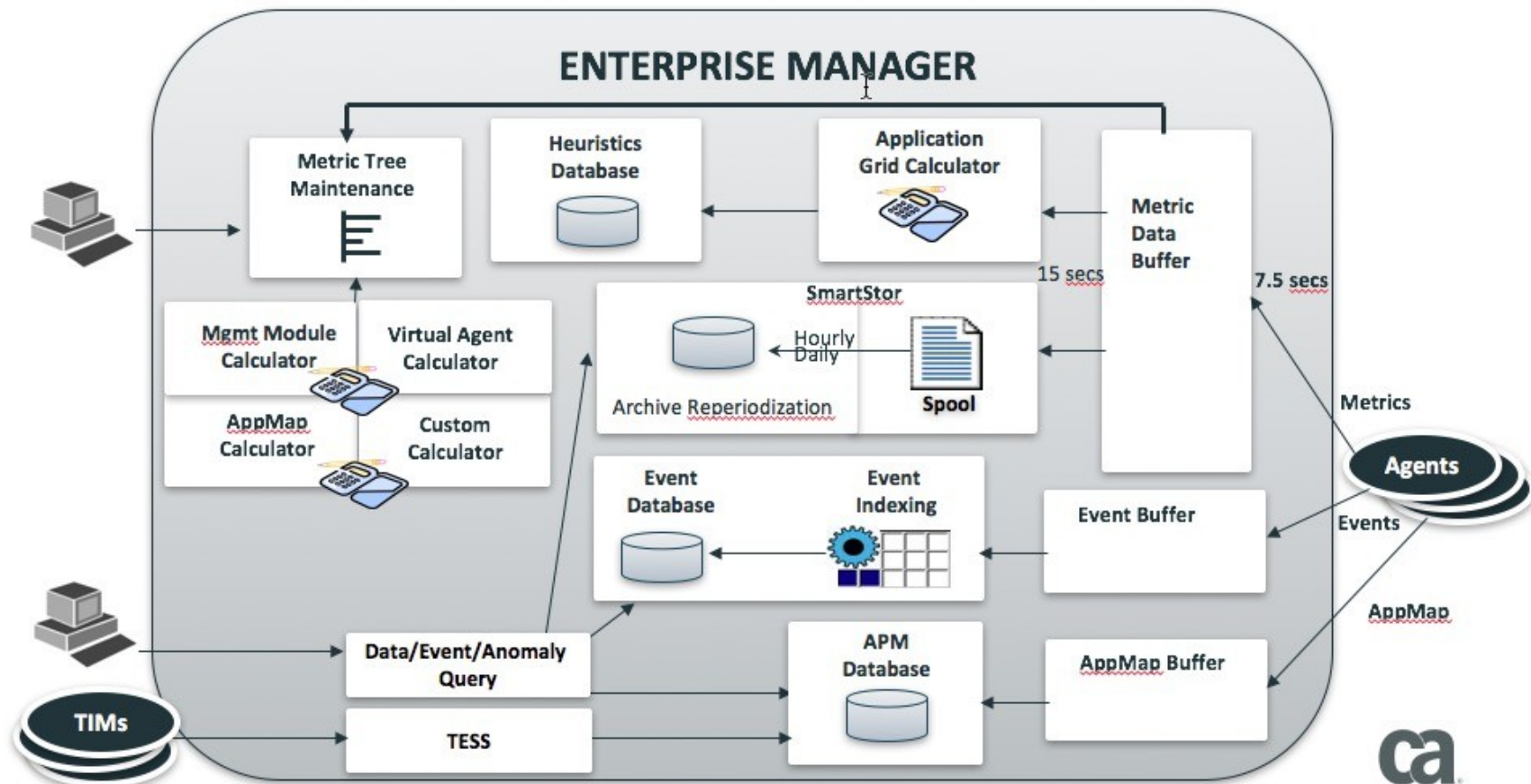
# Déploiement: Extraction donnée

- Extractions de données
  - Définir les requêtes CLW/APMSQL + tester.  
En même temps garder les collecteurs/MoM en vue (mémoire, collection – harvest - et SmartStor)
- Créer des Attributs & Perspectives
- Définir les rapport (quotidiens, hebdomadaires, mensuels)
- Faire un bilan de santé regulier  
(voire partie “Heal Check”)

# EM: meilleures pratiques



# EM: architecture interne



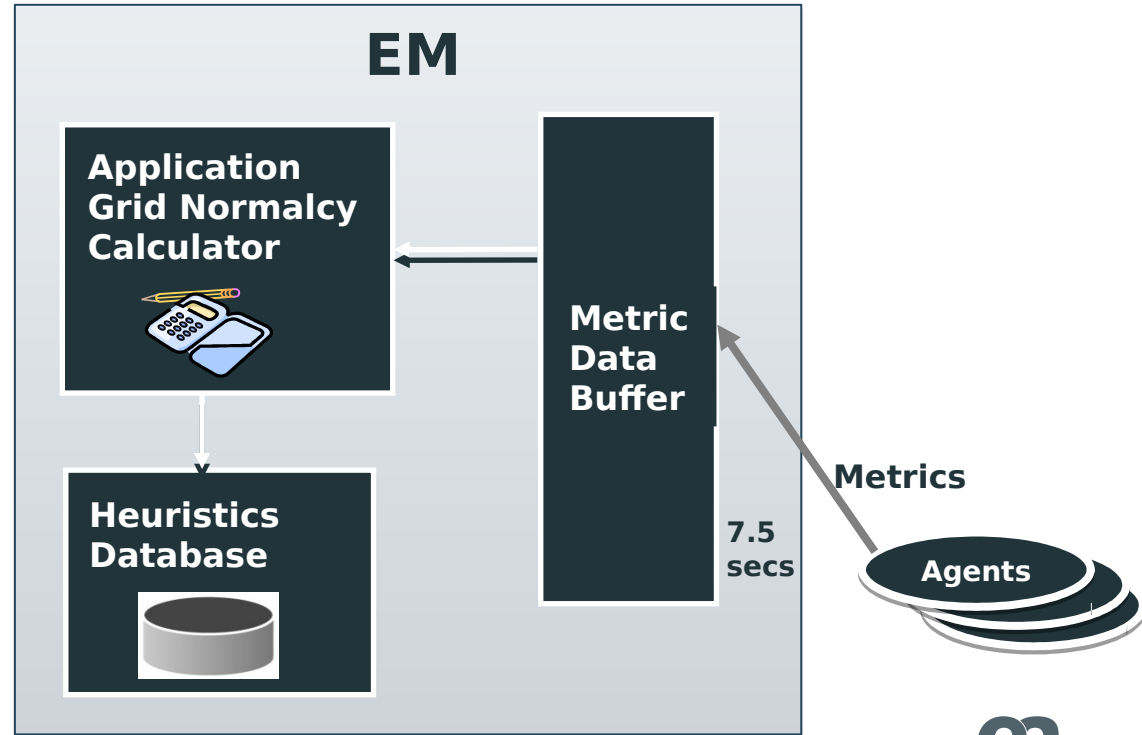
# EM: Processus interne principal - Gestion des événements

## Collection des Agents

- Les données entrantes sont agrégées dans le “tampon” de métrique

## Cycle de récolte

- Les données du tampon métrique sont collectées et envoyées aux abonnés
- Utilise un mécanisme de verrouillage (Locking)





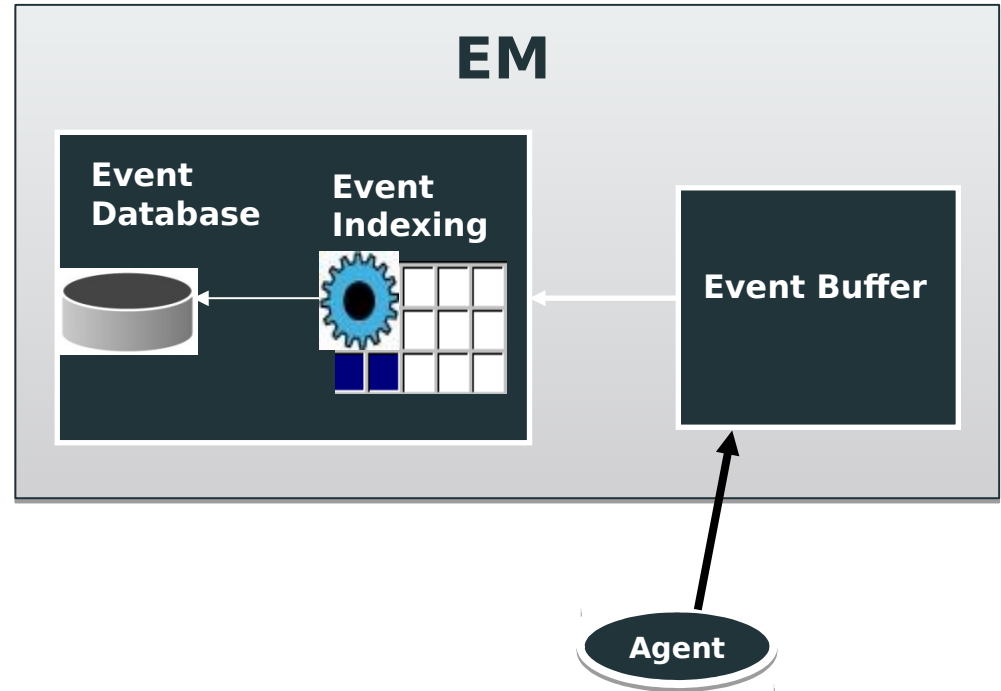
# EM: Processus interne principal - Gestion des événements

## Le stockage des événements

- Initialement stockés dans un tampon
- Détails des événements sont indexés pour permettre des recherches historiques
- Sauvegardé sur disque

## Les événements consistent de

- Traces des transactions
- photo instantanée des erreurs/stalls
- Choses intéressantes



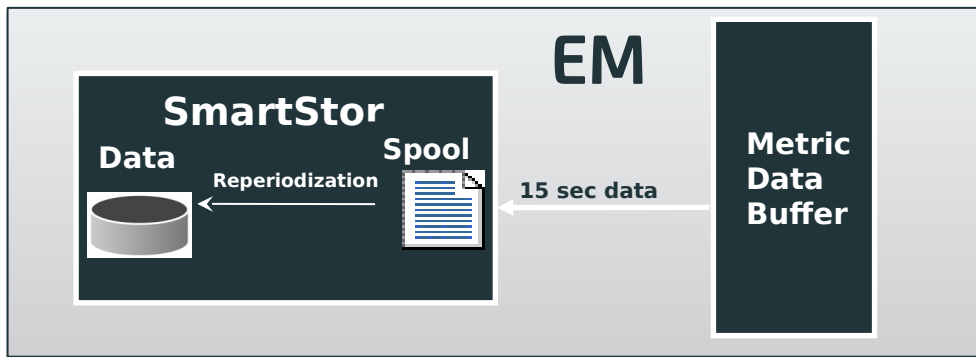
# EM: Processus interne principal - Données SmartStor

## SmartStor Spooling

- SmartStor écrit dans son format spool

## SmartStor Other Tasks

- Converti spool en données
- Compartimenter/divise
- Ajout/insertion (APM 10.5.x) de fichiers de données



# EM: Processus interne principal - fonctions de SmartStor

Toutes les 15 Seconds

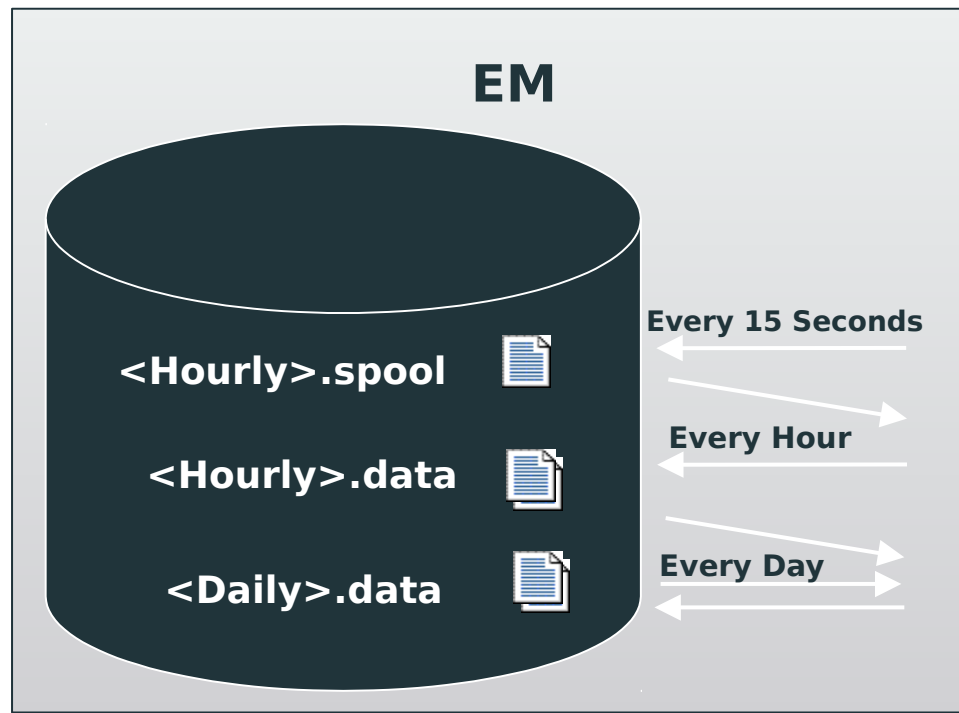
- Converti les données métriques au format \*.spool

Chaque heure

- Convertit les données \*.spool de la dernière heure au format \*.data

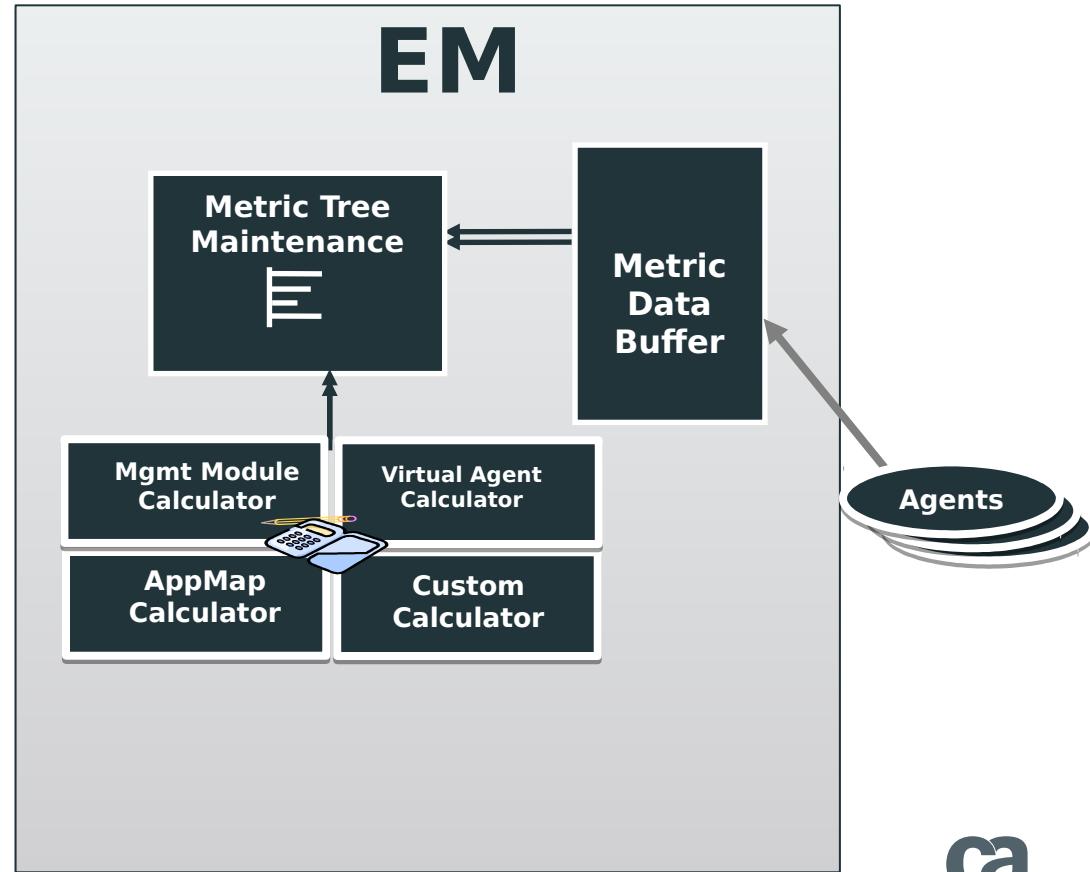
Chaque jour

- Fusionne 24 fichier de 1 heure du jour \*.data en 1 fichier quotidien \*.data
- Agrégation des métriques
  - Du niveau 1 au niveau 2
  - Du niveau 2 au niveau 3



# EM: Processus interne principal - Présentation des données métriques

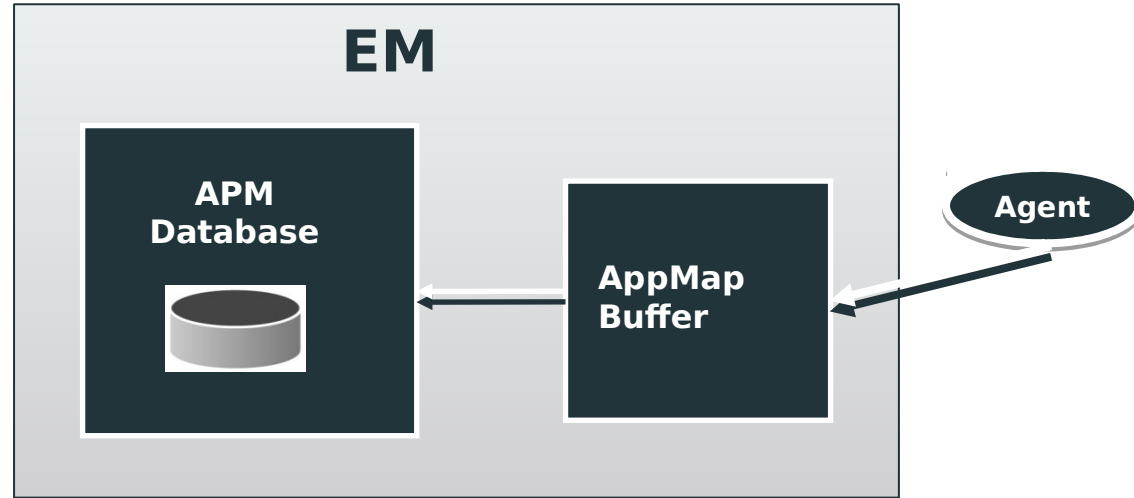
- Présentation des données aux clients
- Tâches
  - Calcul d'agent virtuel
  - Calcul du module de gestion
    - Seuils d'alerte
    - Actions de script
  - Entretien des arbres métriques
  - Calculateurs récapitulatifs (Summary calculators)
    - AppMap
      - Frontend
      - Backend
    - Business Service
      - Customer Experience
      - Business Segment



# EM: Processus interne principal – App Map (Carte D'application)

## Stockage AppMap

- L'information des limites (Edges) et Vertex à sauvegarder dans APM DB
- Les vertex identifient les noeuds pour la AppMap et les composantes FrontEnd/BackEnd
- Les limites (Edge) identifient la relation entre le frontend et le backend



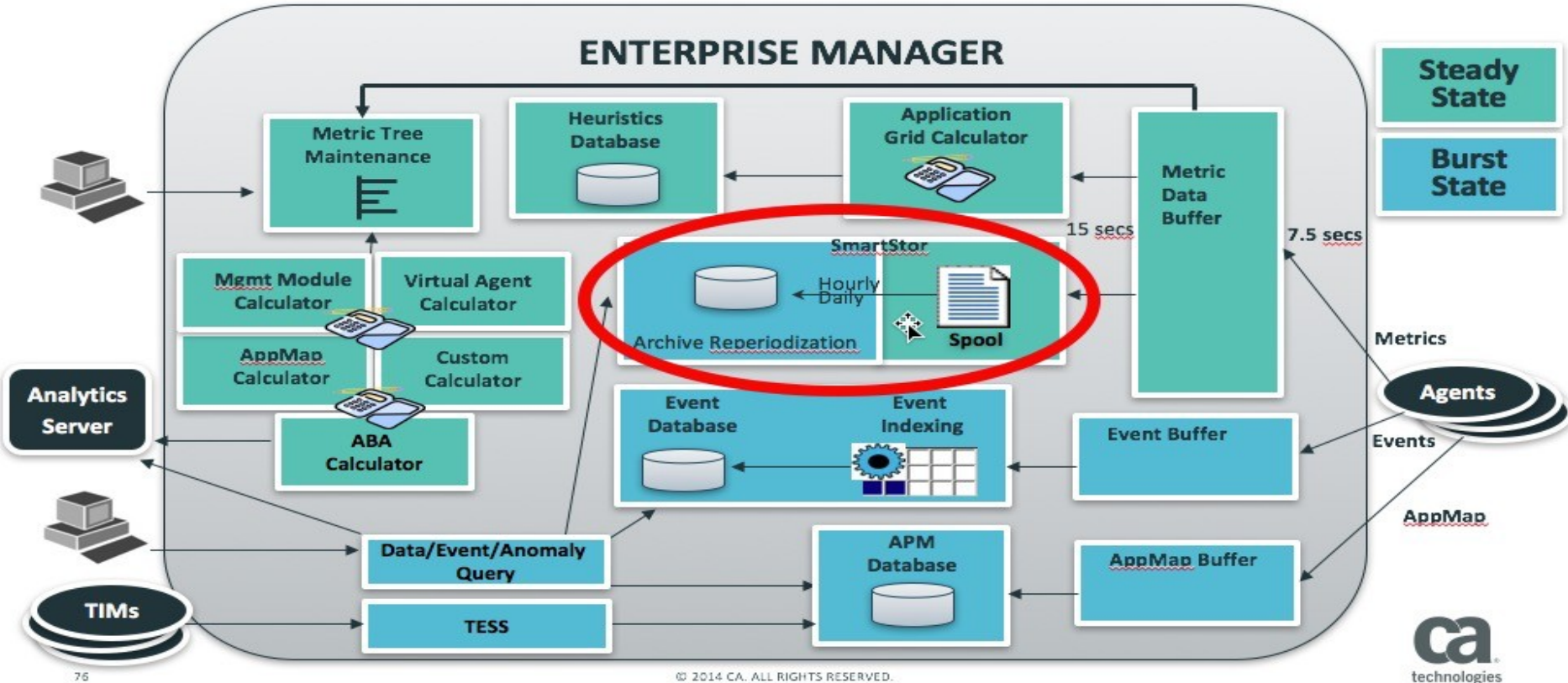
# EM: Guide de dimensionnement

- 2000 métriques/agent
- < 400 agents/collector
- 400K métriques/collector
- 3.0M métriques/collector
- < 5 M métriques par cluster
- collecte, durée Smart Store < 3.5s
- Durée GC < 1 to 1.5 Sec
- Utilisation CPU < 55 to 65 %

# EM: Overheads

- Explosion de métriques (Metric Explosion)
- Fuite de métriques (Metric Leak)
- Équilibrage de charge fréquent
- Trop de traitement (Calculators, Dashboards etc)
- Trop de récupération de données
- Systèmes sous dimensionnés (CPU, mémoire ...)

# EM: Fuite de métriques – un exemple





# EM: Fuite de métriques – cause/impacte

## Causes les plus communes

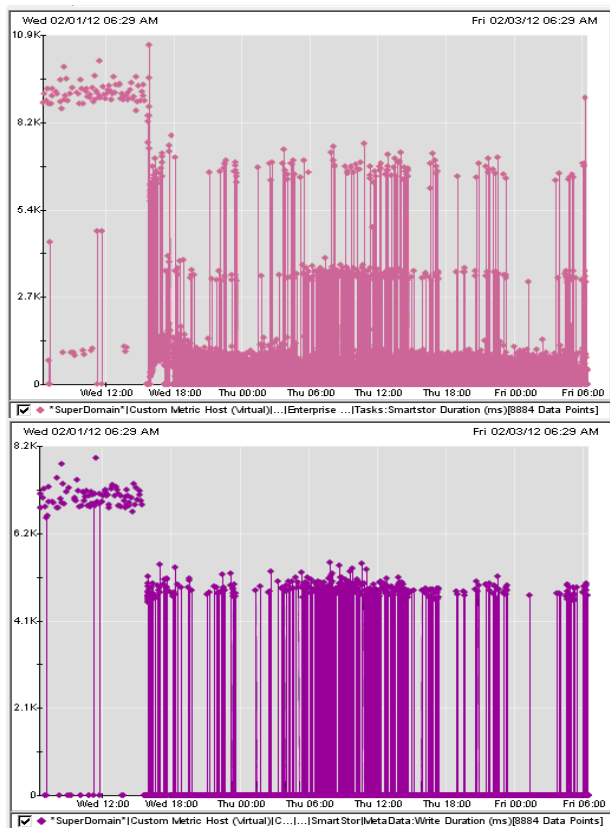
- Création constante de nouveaux nom de métriques (p.e. classes temporaires)
- Mauvais formatage de nom (Bad name formatter - report SQL values)

## Impact visible

- Large “Smartstor Duration (ms)” couplé avec une “Write Duration (ms)” large + augmentation du nombre de métriques historiques → augmente le temps entre les collecte des métriques, réduit le nombre de E/S (IO bottleneck) etc.
- Traitement retardé “Number of Metrics Handled”
- “Ping Time (ms)” élevé entre MOM et les Collector lents
- Les collecteurs vont se déconnecter
- Interaction avec la Workstation très lente

# EM: Fuite de métriques

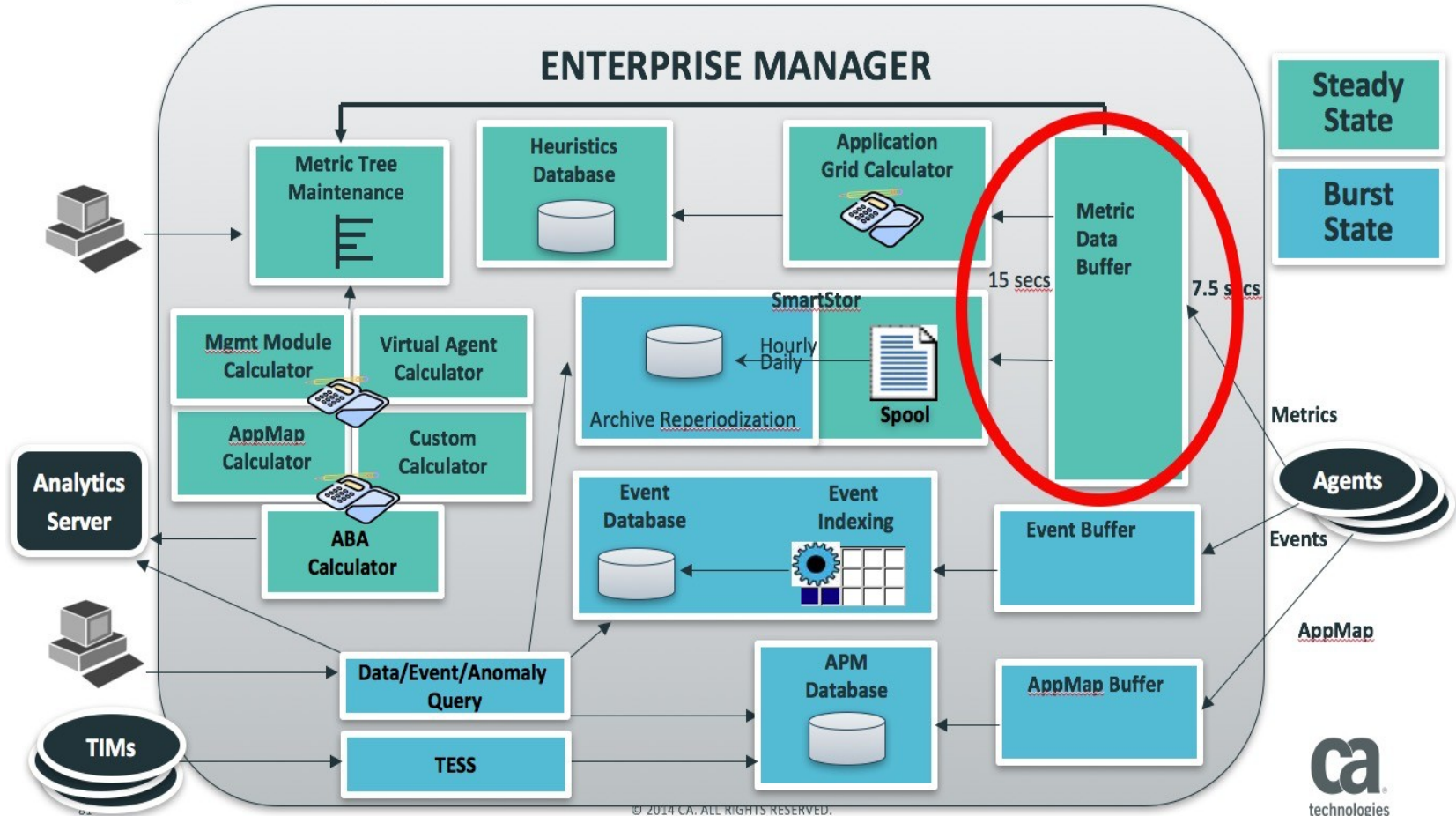
SmartStor Duration (ms)  
couplé avec  
Write Duration (ms)



**SmartStor Duration (ms)**

**Write Duration (ms)**

# EM: Explosion de métriques – un exemple



# EM: Explosion de métriques – cause/impacte

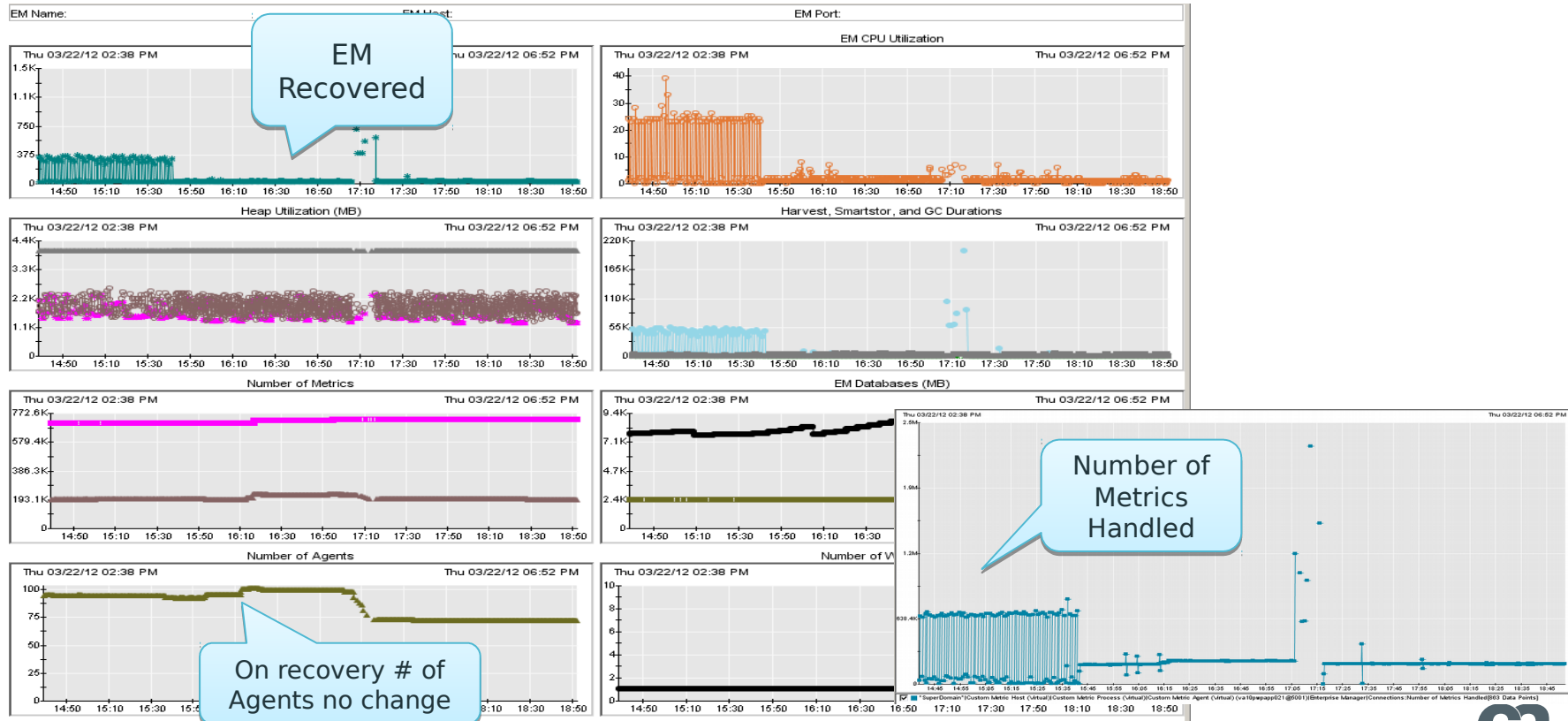
## Causes les plus communes

- Agents livrant beaucoup de métriques

## Impact visible

- “Harvest Duration (ms)” élevée; Métriques de tous les agents.
- “Agent 50K metric clamp reached” dans les logs
- Traitement retardé “Number of Metrics Handled”
- “Ping Time (ms)” élevé entre MOM et les Collector lents
- Les collecteurs vont se déconnecter
- Interaction avec la Workstation très lente

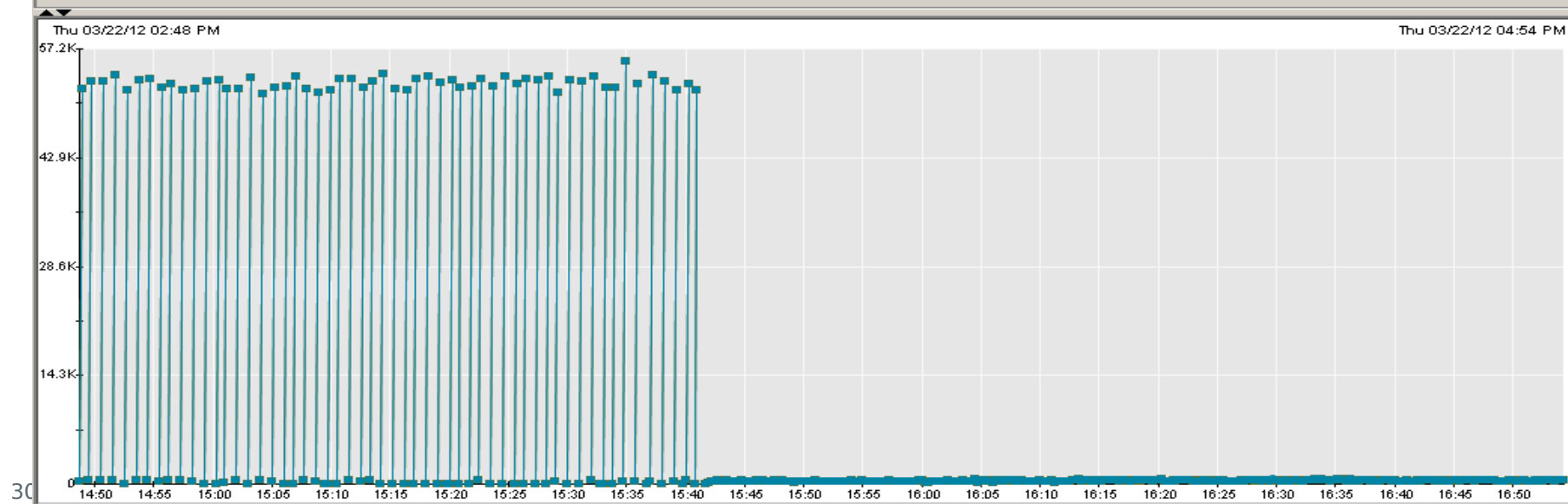
# EM: Explosion de métriques – impacte d'un mauvais agents sur le EM



# EM: Explosion de métriques – temp de collecte d'un agent

- Agent Harvest thread CPU (active processing) and User (total processing) times are high and identical. Majority time processing incoming agent data storm
- [Manager.Clock] Time to harvest agents: 52883ms

Resource	Metric	Value	Min	Max ▾	Count
Enterprise Manager\Internal\Threads\PO WatchedAgent\PO Mailman	Wait Time (ms)	462,625	2,610	462,625	408
Enterprise Manager\Internal\Threads\PO WatchedAgent\PO Mailman	Blocked Time (ms)	80,447	0	80,447	408
Enterprise Manager\Internal\Threads\PO WatchedAgent\PO Mailman	User Time (ms)	56,941	0	56,941	408
Enterprise Manager\Internal\Threads\PO WatchedAgent\PO Mailman	CPU Time (ms)	56,940	0	56,940	408



# Agent: meilleures pratiques

- Avec APM 10, un nouvel outil d'administration des agents a été mis à disposition:

APM Command Center (ACC).

- Si vous avez des application surveillé par l'agent APM 10+, il est fortement recommandé d'utiliser ACC pour la gestion des agents

# Agent: Installation / Fonctionnement

- Définir les rôles (à faire une fois)
- Définir une configuration standard adapté à l'environnement pour les différents agents WAS, Tomcat etc. (à faire une fois)
  - 1) Identifier l'application
  - 2) Définir un plan de projet
  - 3) Identifier le Cluster APM (pour connecter l'agent)
  - 4) Définir les Baselines (fonctionnement normal)
  - 5) Déploiement (QA, Pre-Prod, Prod)
  - 6) Verifier le tout
- Inclure APM dans la “politique” de fonctionnement des équipes



# Agent: Configuration / Detail

## Exemple de configuration de base pour l'agent

Desactiver	Socket, NIO, JMX, EntityBean, Leak Hunter, Change Detector, individual SQL
Limiter (clamp)	Live Metric, Excessive Errors and TT
propriétés	<code>Introscope.agent.sqlagent.sql.maxlength=xxx</code> where xxx between 1 and 990, remove {sql} from pbd, <code>introscope.autoprobe.dynamicinstrument.enabled=false</code> (for prod), <code>introscope.agent.transactiontrace.componentCountClamp</code> (lower if too many incoming TT of type "normal"), ACC is enabled and turned on

Sauvegarder les informations comme JVM/.Net Version, Framework (Struts, MVC etc) pour créer une référence pour usage future.

Note: Le nouveau ACC va mettre ces informations à disposition.

# Agent: Fonctionnement / Detail

- Surveiller des Conteneur et des Microservices
  - Surveiller l'application executé dans le conteneur avec l'agent APM
    - Définir une convention pour les noms: conteneur, host, process et agent. Exemple: ServiceName, AppName, ProjectName
    - Du coté de l'agent et du EM, le vieillissement et la purge de métrique peut être utilisé pour éliminer les métrique et agents mortes et les agent
    - Désactiver l'agent qui surveille la plate-forme
- Surveiller le conteneur et la topologie des clusters de conteneurs
  - Désactiver les métriques de l'agent plate-form, utiliser l'extension Docker
  - Utiliser l'extension APM SysDig si APM 10.5.1 agent est utilisé

# Agent: Pre-installation

- Identifier les application nécessitant un agent
  - En fonction de la priorité
  - Commencer avec une ou deux applications
- Définir un plan de projet en fonction de
  - Chronologie, ressources, dépendances, risques, etc.
- Identifier le cluster de production
  - Décision basée sur la capacité, l'emplacement etc.

# Agent: Installation

- Définir des Baselines en utilisant la configuration “standard”
  - Vérifier configuration, application et performance avec l’ACC
  - Définir/créer les Dashboards, Alertes
  - Tester la configuration
- Déploiement
  - Utiliser l’ACC pour installer les packets
  - Déployer les agents en Phases avec QA, Pre-Prod, Prod etc
  - Chercher des mauvaise(s) configuration ( Metric Leak, Explosion etc.)
  - Utiliser les résultats pour adapter les Baselines

# Agent: Post-installation

- Retour d'information après installation
  - QA, développeurs, utilisateurs
- Faire un bilan de santé régulier de l'installation
  - Vérifier les Dashboard des MOM/Collector et confirmer que principaux indicateurs de performance comme GC, SS, Harvest, CPU etc sont Ok
- Définir le cycle de vie du développement logiciel en tenant compte de APM

# Agent: bilan de santé régulier

- Vérifier la performance de l'application régulièrement (hebdomadaire)
- Vérifier les Dashboard des MOM/Collector:  
mettre l'intervalle à 6heures et la granularité à 15secondes
- Rechercher des anomalie comme une valeur élevée pour le SmartStor, Harvest, GC, CPU, Historical, Live metrics, Agent Disconnect, AppMap, DA etc.
- Des pics occasionnels sont acceptables, des pics récurrents sont à vérifier