

5^{ème} année SDBD

Mineure

Software Defined Communication Infrastructure

Projet SDCI

Lien d'accès : tiny.cc/Projet-SDCI

Equipe enseignante :

S. Medjiah, S. Yangui, C. Ouedraogo, EF. Bonfoh, C. Chassot

Plan de la présentation

- Compétences ciblées
- Modalités de travail et d'évaluation
- Objectifs, pré requis et séquençement du projet
- Présentation de l'activité IoT ciblée et de son évolution en 3 phases
- Vision "IT" de l'activité ciblée et de son évolution
- Travail demandé et organisation
- Plateforme et outils mis à disposition
- Méthodologie de conception

Compétences ciblées
+
Modalités de travail et d'évaluation

Compétences ciblées (1/3)

A l'issue du projet, l'étudiant sera capable de :

- En termes de savoir :
 - Connaître les solutions d'interconnexion des applications logicielles et des capteurs / actionneurs impliqués dans l'IoT, par le biais d'un middleware distribué
 - Comprendre les points de vue des ≠ acteurs du monde réel impliqués dans un projet de ce type
 - Développeur d'application IoT, Opérateur middleware (3d party), Opérateur réseau
 - Comprendre les concepts attendant à la virtualisation de fonctions de réseau dans un système de communication basé sur un / des réseaux programmables (au sens : définis par le logiciel)
 - Connaître le modèle de l'autonomic computing défini par IBM

Compétences ciblées (2/3)

A l'issue du projet, l'étudiant sera capable de :

- En termes de savoir-faire :
 - utiliser un émulateur de réseau SDN = **ContainterNET** (incluant la description d'une topologie réseau via le langage python) (NB : containernet = mininet++)
 - utiliser un contrôleur SDN (**Ryu**)
 - utiliser un MANO NFV standardisé (**SON-EMU**)
 - développer une VNF standardisée ETSI
 - invoquer une interface REST
 - utiliser (voire développer) une application IoT sur la base d'un middleware distribué
 - ...

Compétences ciblées (3/3)

A l'issue du projet, l'étudiant sera capable de :

- En termes de compétence :
 - Architecturer et mettre en oeuvre des solutions tirant partie des concepts de virtualisation de fonctions de réseau (au sens NFV) et de réseaux pilotables par le logiciel (au sens SDN), dans le contexte de la réalisation d'une SDCI
 - Appliquer et mettre en oeuvre des éléments du modèle de l'autonomic computing en réponse à une problématique de gestion dynamique de QoS dans une SDCI

Modalités de travail et d'évaluation

- Travail par groupe de 2 étudiants en TD et en TP
- Evaluation sous la forme :
 - Rapport final
 - Soutenance finale incluant démonstration
 - Soutenance intermédiaire
 - Implication individuelle durant les TD / TP

Objectif, pré-requis et séquençement du projet

Objectif, pré-requis et séquençement du projet (1/3)

- **Double objectif**

- Déployer *dynamiquement* et de façon transparente des fonctions de réseau virtuelles (VNF)
 - permettant de répondre aux *besoins fonctionnels* et/ou *non fonctionnels d'applications distribuées* relevant d'une activité de l'Internet des objets (IoT)
 - en appliquant les concepts et techniques relevant de la *virtualisation de fonctions de réseau (NFV)* et des *réseaux pilotables par le logiciel (SDN)*
- Développer une approche de *gestion autonome* de la mise en œuvre des VNF ciblées
 - via le concept de l'*Autonomic Computing (AC)* introduit en préambule du projet

Objectif, pré-requis et séquençement du projet (2/3)

- **Pré requis**

- Autonomic computing ⇒ 5SDBD
- Virtualisation de fonctions de réseaux (NFV et SDN) ⇒ 5SDBD
- Concepts et techniques logicielles associés à la virtualisation ⇒ 5SDBD
- Virtualisation de réseaux ⇒ 5SDBD
- Interconnexion de réseaux ⇒ 4IR
- Introduction aux réseaux ⇒ 3MIC

+

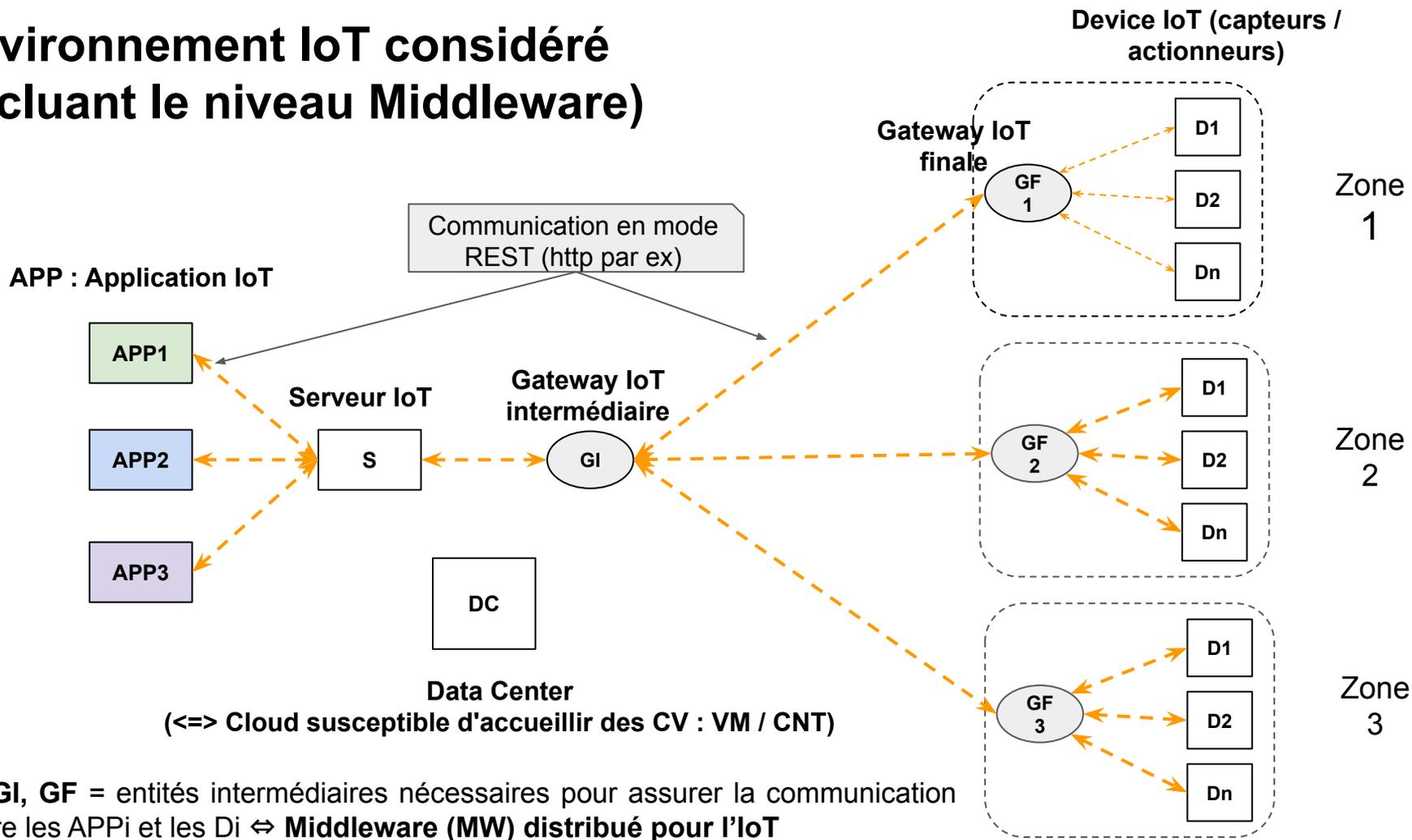
- Architecture logicielles orientées services ⇒ 5SDBD
- Conception orientée objets ⇒ 4IR
- Langage de programmation orientée objet (JAVA, ...) ⇒ 4IR

Objectif, pré-requis et séquençement du projet (3/3)

- **Séquençement prévisionnel du projet (susceptible d'être amendé)**
 - 2 séances CM : Présentation du projet + Autonomic Computing
 - 1 séance TD : Conception
 - 1 séance TP (4h) : Mise en place des outils
 - 2 séances TD : Conception
 - 1 séance TP (4h) : Développement
 - 1 séance TD : Conception
 - Soutenance intermédiaire : Présentation de la conception par chaque binôme
 - 4 séances TP (4h chacune) : Développement
 - Soutenance finale, incluant démonstration
 - Débriefing entre enseignants et étudiants

Présentation de l'activité IoT ciblée et de son évolution en 3 phases

Environnement IoT considéré (incluant le niveau Middleware)

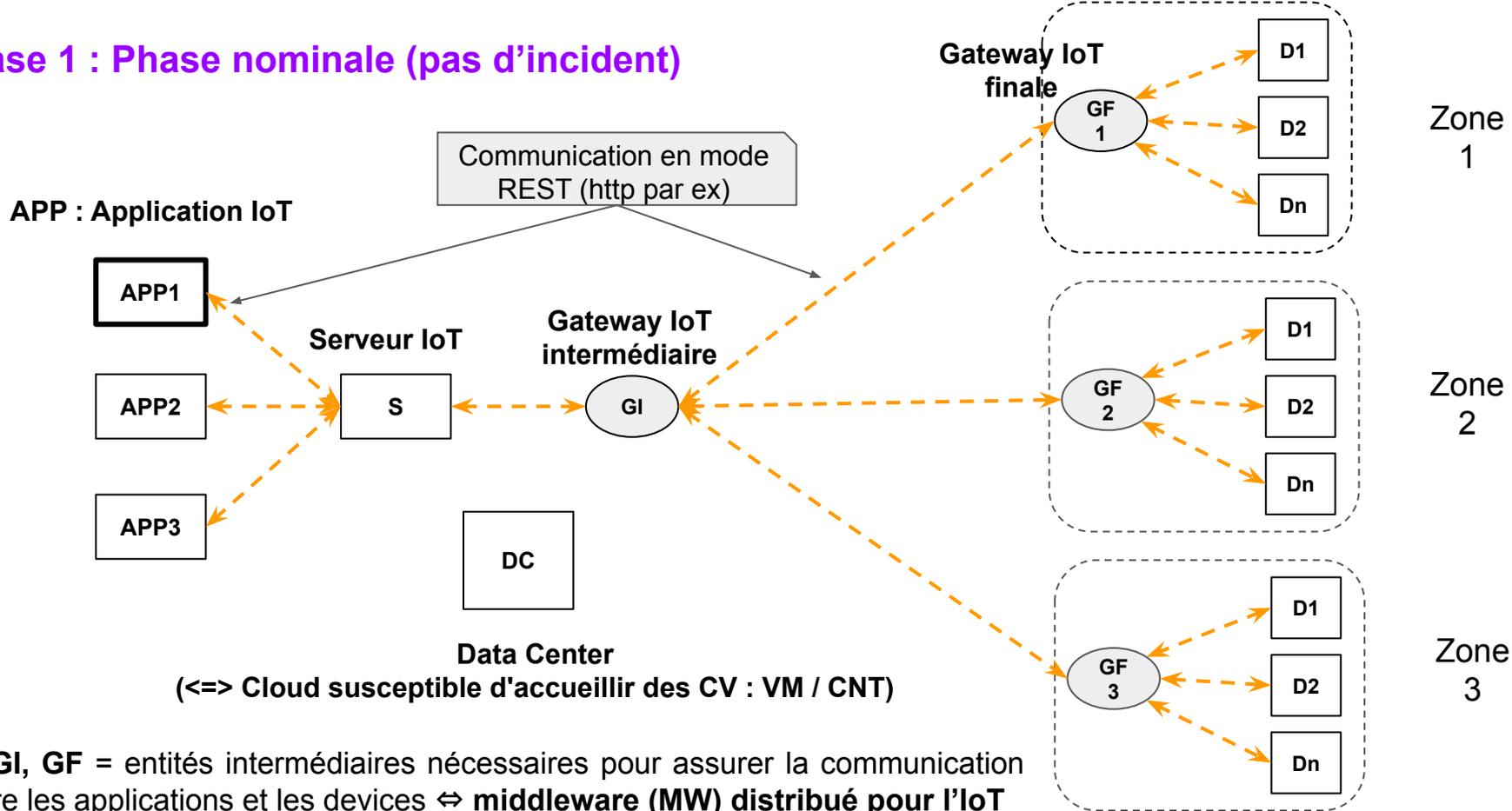


S, GI, GF = entités intermédiaires nécessaires pour assurer la communication entre les APPi et les Di ⇔ **Middleware (MW) distribué pour l'IoT**

Activité IoT ciblée : Activité de supervision/intervention à distance de ≠ zones dotées de capteurs / actionneurs (Di), par le biais d'une application (APP1)

Device IoT (capteurs / actionneurs)

Phase 1 : Phase nominale (pas d'incident)

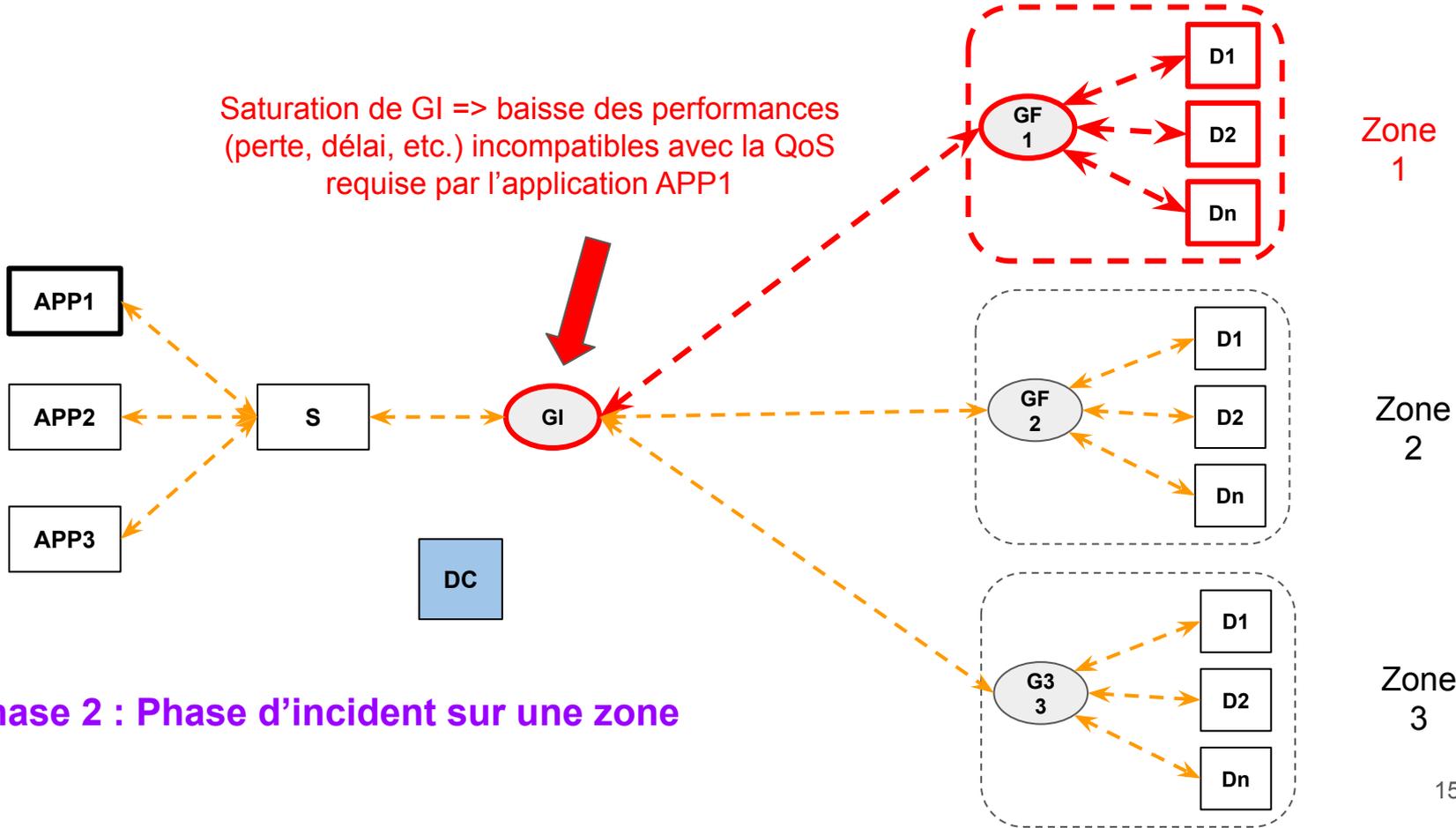


S, GI, GF = entités intermédiaires nécessaires pour assurer la communication entre les applications et les devices ⇔ **middleware (MW) distribué pour l'IoT**

Activité IoT ciblée : Activité de supervision/intervention à distance de ≠ zones dotées de capteurs / actionneurs (Di), par le biais d'une application (APP1)

Incident sur la zone 1 => trafic supplémentaire généré par les Di

Saturation de GI => baisse des performances (perte, délai, etc.) incompatibles avec la QoS requise par l'application APP1



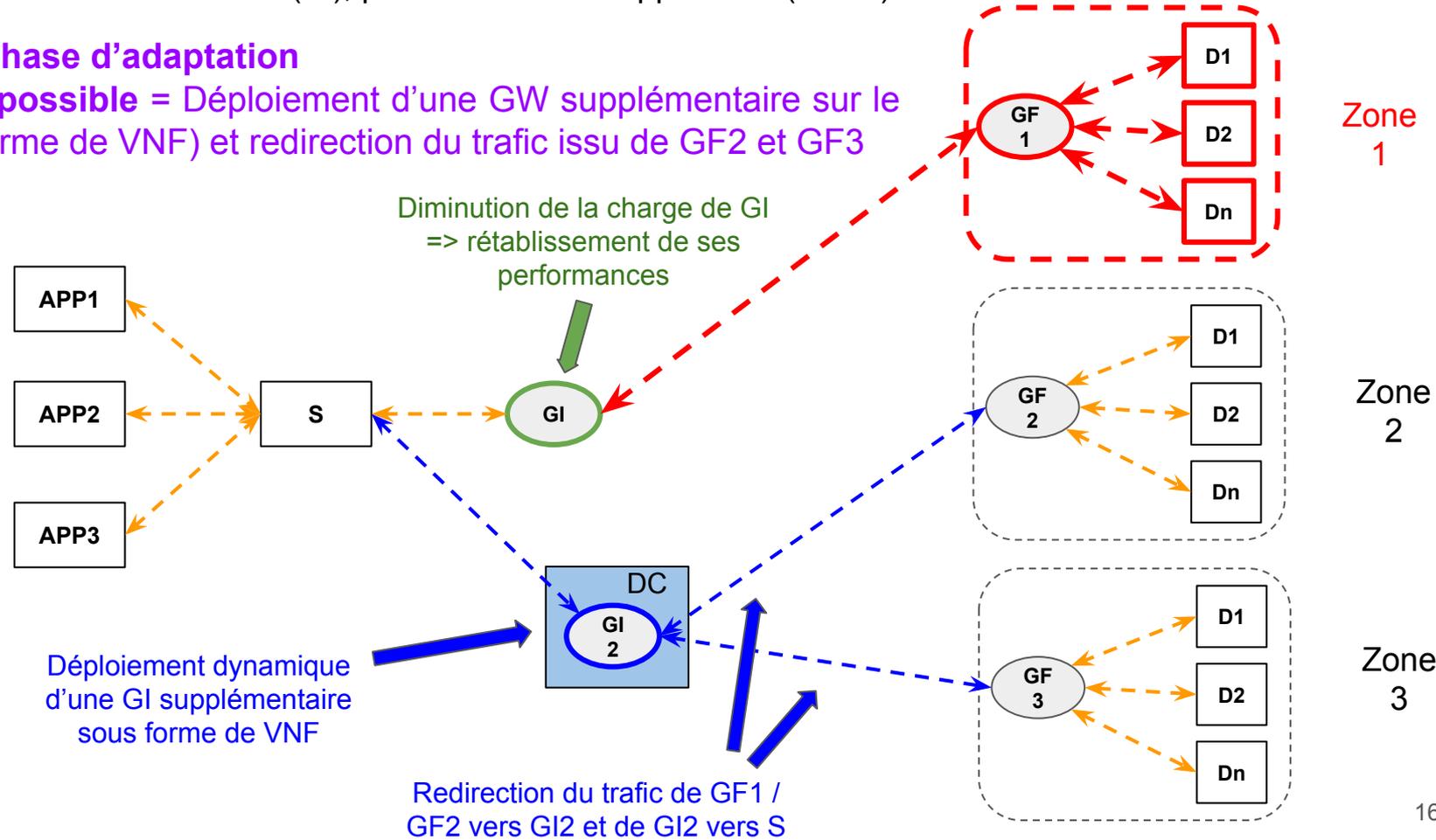
Phase 2 : Phase d'incident sur une zone

Activité IoT ciblée : Activité de supervision/intervention à distance de ≠ zones dotées de capteurs / actionneurs (Di), par le biais d'une application (APP1)

Incident => trafic supplémentaire générés par les Di

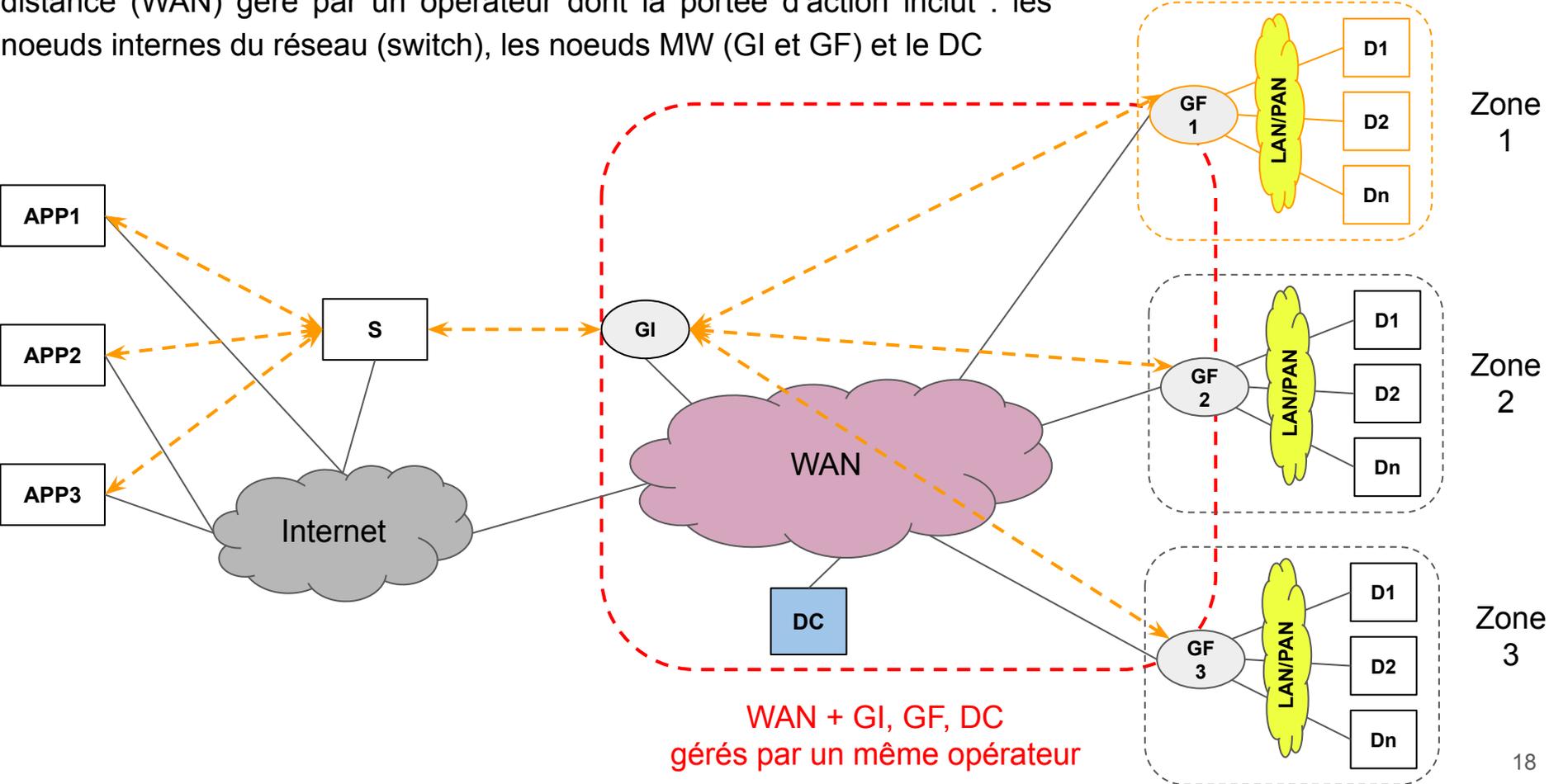
Phase 3 : Phase d'adaptation

1 exemple possible = Déploiement d'une GW supplémentaire sur le DC (sous forme de VNF) et redirection du trafic issu de GF2 et GF3

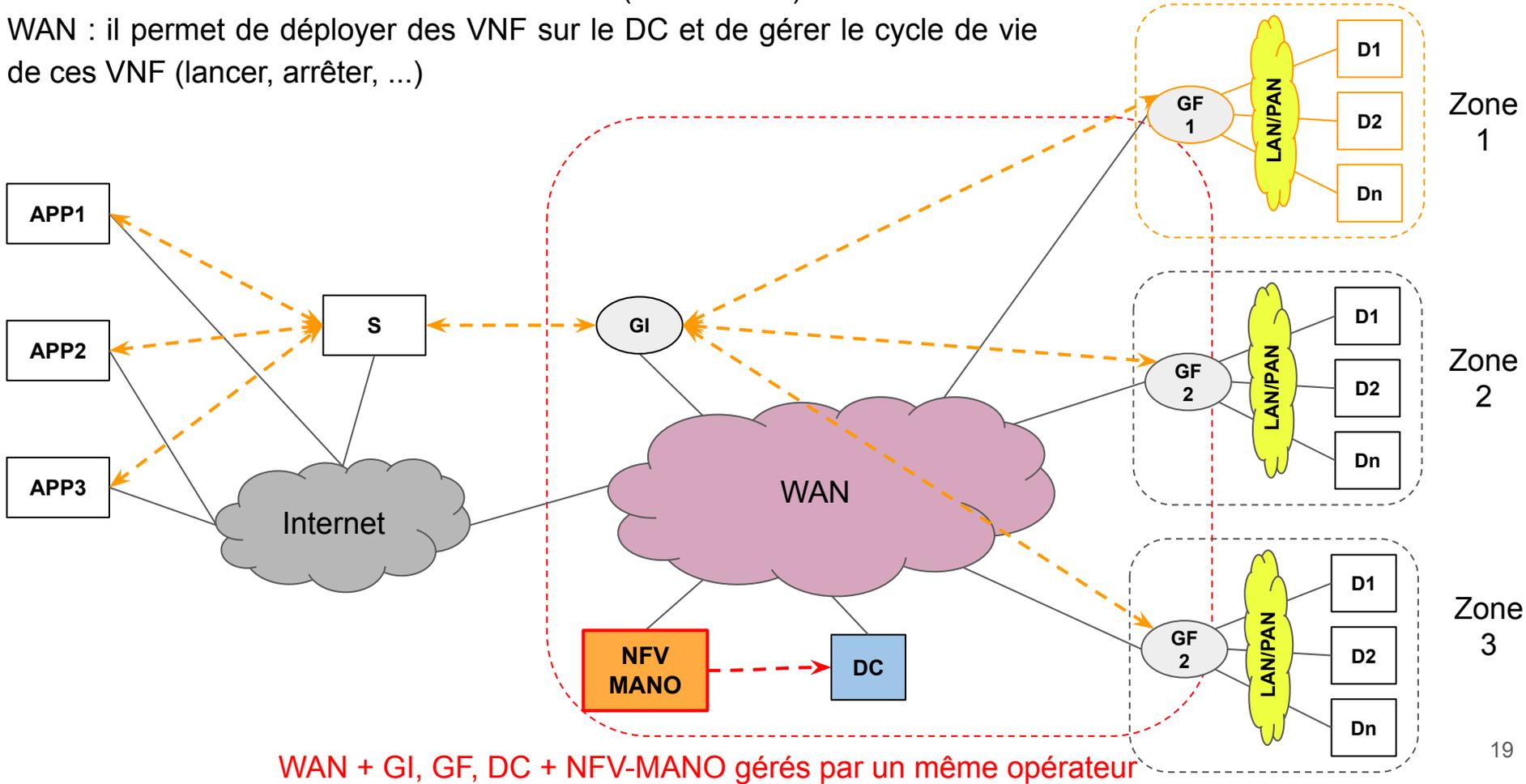


Vision “IT” de l’activité ciblée et de son évolution

HYPOTHÈSE 1 : GI, GF et DC sont connectés via un réseau grande distance (WAN) géré par un opérateur dont la portée d'action inclut : les noeuds internes du réseau (switch), les noeuds MW (GI et GF) et le DC



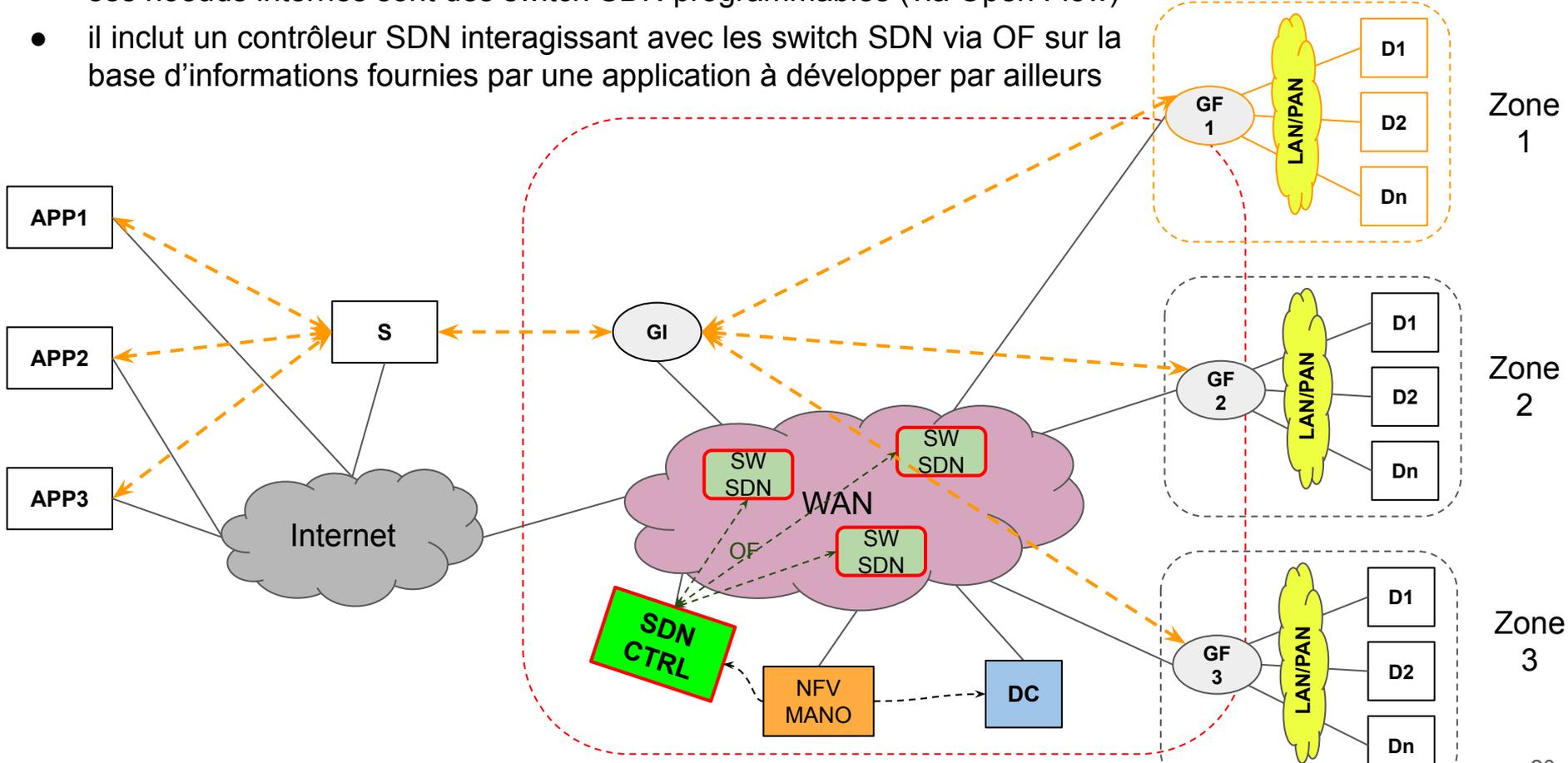
HYPOTHÈSE 2 : Un orchestrateur de VNF (VNF-ORCH) est connecté au WAN : il permet de déployer des VNF sur le DC et de gérer le cycle de vie de ces VNF (lancer, arrêter, ...)



WAN + GI, GF, DC + NFV-MANO gérés par un même opérateur

HYPOTHÈSE 3 : Le WAN est doté de capacités SDN, i.e. :

- ses noeuds internes sont des switch SDN programmables (via Open Flow)
- il inclut un contrôleur SDN interagissant avec les switch SDN via OF sur la base d'informations fournies par une application à développer par ailleurs



WAN + GI, GF, DC + MANO NFV+ SDN-CTRL gérés par un même opérateur

Travail demandé et méthodologie

Travail demandé et organisation (1/2)

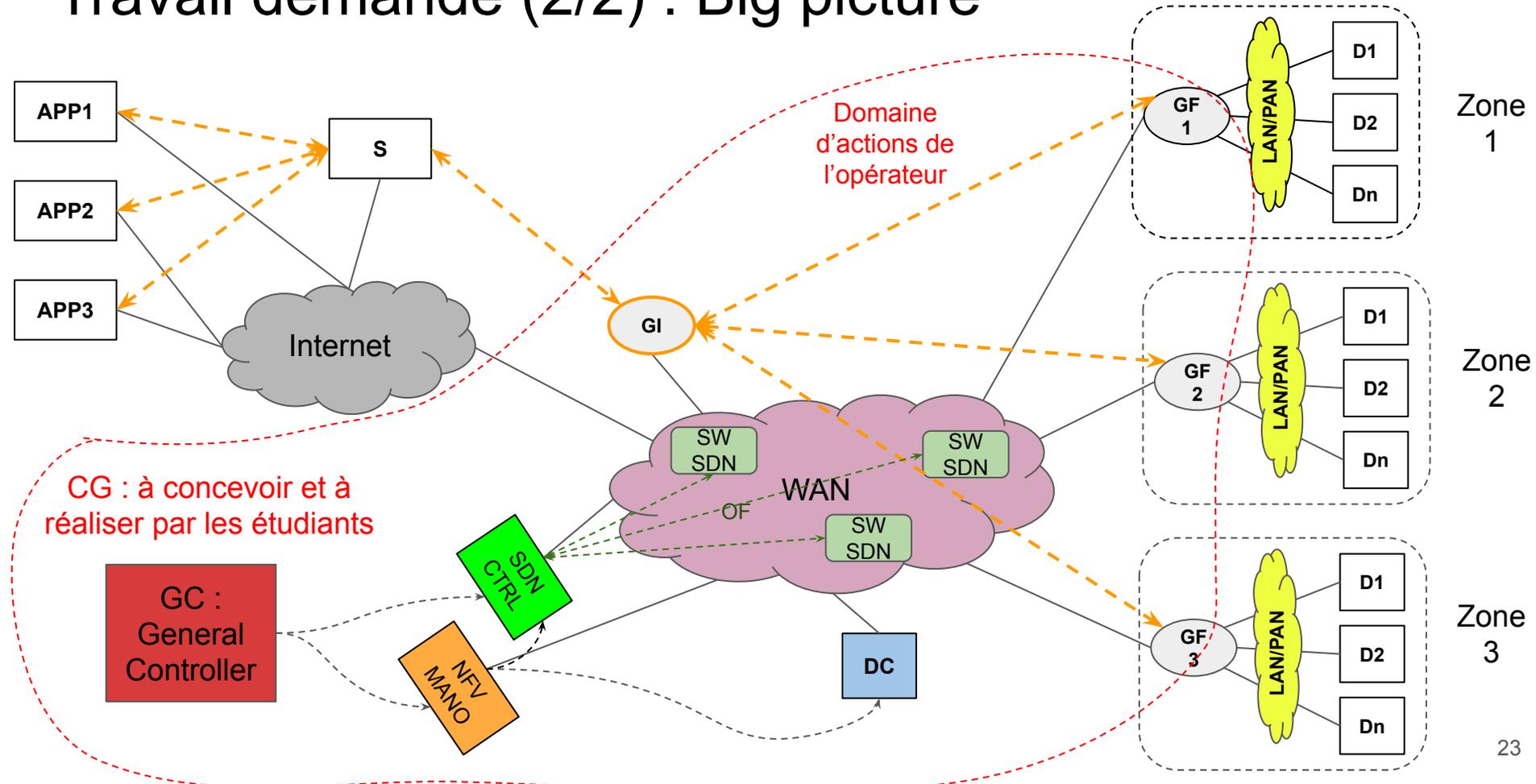
Travail demandé

- Mettre en place l'adaptation requise lorsque l'activité passe de la phase 2 à la phase 3, suivant le cadre de l'Autonomic Computing

Organisation

- Sur la base du cas d'étude présenté, identifier des mécanismes d'adaptation et les détailler sous la forme de use case (UML) simples => objet du TD1
- Proposer une boucle autonome pour l'application de ces mécanismes d'adaptation (Boucle V1)
- Identifier des mécanismes supplémentaires (adaptation ou observation) afin d'enrichir la boucle autonome (Boucle V2)
- Identifier des mécanismes supplémentaires pour améliorer l'analyse et/ou la planification de la boucle autonome (Boucle V3)

Travail demandé (2/2) : Big picture



Plateforme et outils mis à disposition

Plateforme et outils mis à disposition (1/2)

PF d'émulation de réseau ContainerNet (\Leftrightarrow mininet++) (sous forme de VM) incluant :

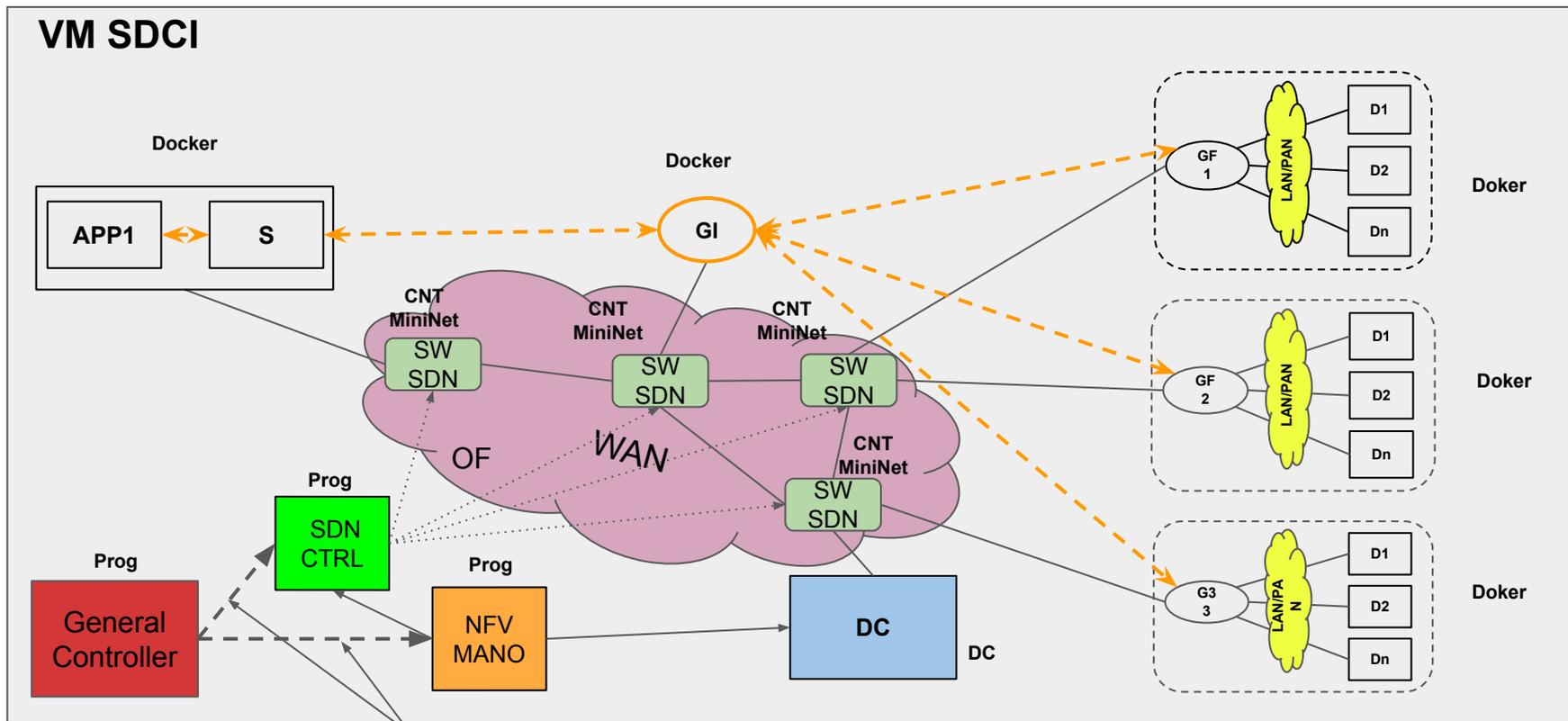
- le programme mn (vu en TP SDN), permettant :
 - d'émuler une topologie de réseau SDN (switchs et hôtes) sous forme de containers LXC ou **Docker**
 - de connecter les switchs à un contrôleur SDN spécifié lors du lancement
- le contrôleur SDN = **Ryu**
 - permettant de contrôler les switchs créés par mn via le protocole OF
 - exposant une API REST pour développer des applications de contrôle
- un MANO standardisé ETSI NFV: **ETSI OSM + son-emu**
 - permettant (via une API REST ou CLI exposées):
 - de déployer des NS et des VNFs en des endroits donnés
 - de lancer / arrêter / (re)configurer ces NS / VNF, ...
 - ...

Plateforme et outils mis à disposition (2/2)

Middleware IoT/M2M = MW en NodeJS (fourni)

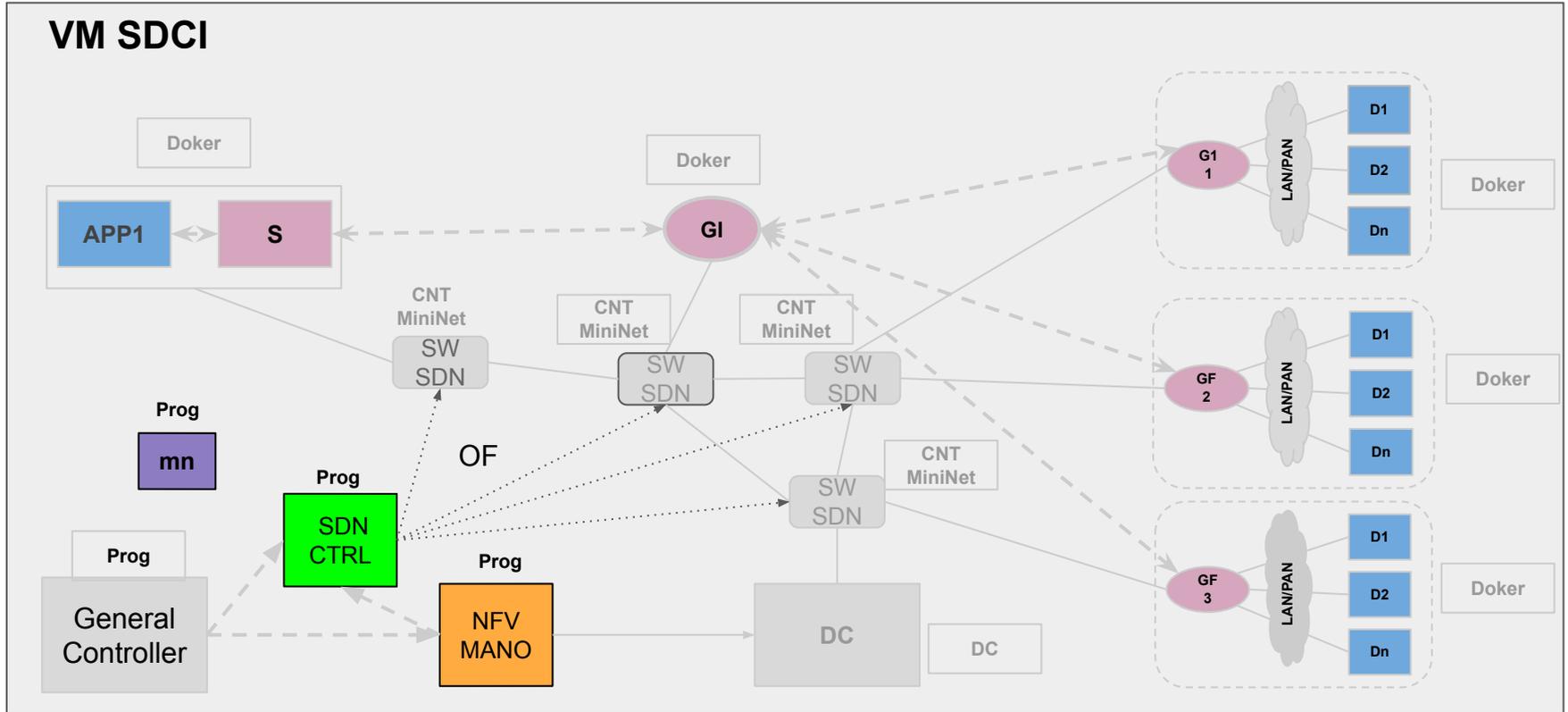
- le programme serveur : **iot-server.js**
 - et sa configuration
- le programme gateway : **iot-gateway.js**
 - et la configuration de chaque gateway (finale et intermédiaire)
- un exemple d'une application (en NodeJS) pour émuler un **capteur**
- un exemple d'une application (en NodeJS) pour exécuter les phases de supervision et d'intervention ↔ sous-ensemble de APP1 (à étendre au besoin pour nourrir un scénario complexe)

Big picture des choix d'implantation



Communication en mode REST (http) (ou en CLI pour le MANO)

Ce qui est mis à disposition des étudiants vs. ce qui est à développer



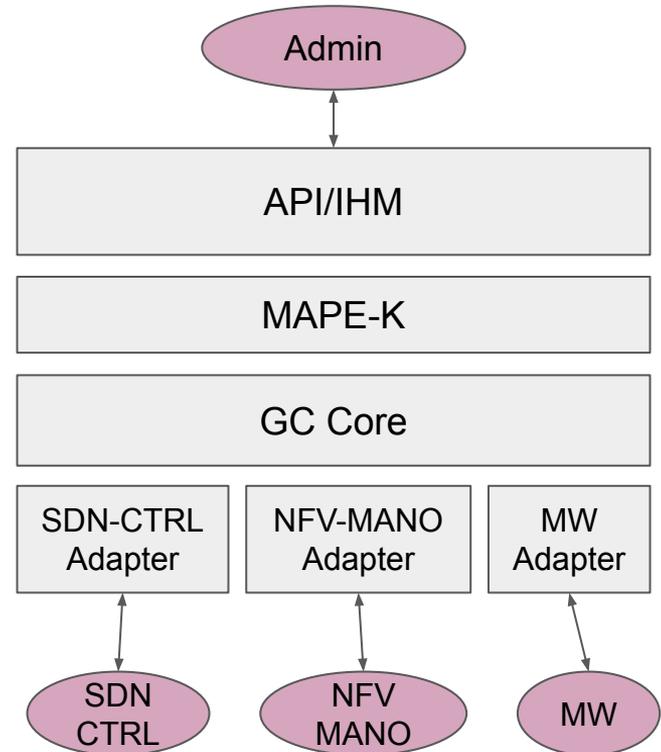
ContainerNet = programme MiniNet++ (pour la création des noeuds, leur interconnexion, etc.)

Méthodologie de conception du CG
⇒ basée UML (cf cours COO de 4IR)

Travail demandé = définir

- Entités (Acteurs) en interaction avec le GC
 - Administrateur
 - SDN-CTRL, NFV-MANO, noeuds MW (GI, ...)
- Use case (à décrire suivant le canevas fourni)
 - UC 1 :
 - UC 2 :
 - ...
- Diagrammes de structure composite
- Diagrammes de séquence
- Diagrammes de classe (JAVA)

Modèle général du GC (à compléter suivant les UC retenus)



Canevas d'élaboration d'un Use Case

- Use case:
 - Identifiant : *USE_CASE_XXX*
 - Version : *X.Y*
 - Nom : (...)
 - Acteurs impliqués : *acteur-1, acteur-2, ..., acteur-N*
 - Description: (...)
 - Objectif visé: (...)
 - Déclencheurs du UC (⇔ action du /des acteurs) : (...)
 - Comportement du système en réponse à la demande : (...)
 - Pré-conditions: (...)
 - Post-conditions: (...)

