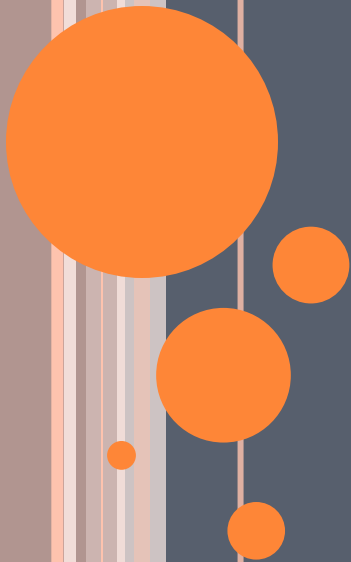


OUTILS DE GÉNIE LOGICIEL

- Cours1 -

Outils et environnements de
développement



OBJECTIF

- Avoir une vue générale sur l'environnement de développement du logiciel
- Voir l'utilité des différents types d'outils utilisés dans le développement de logiciels et leur influence sur l'activité de développement



DÉFINITION :

- En anglais « Software Development Tools »
 - Un outil de développement est tout programme utilisé par les développeurs pour créer, débbugger ou maintenir des programmes et des applications
 - Ces outils sont souvent des programmes relativement simples qui sont combinés pour effectuer les différentes tâches liées aux développement d'une application



HISTOIRE : LA NAISSANCE DES OUTILS

- Les outils de développement existent depuis les premiers ordinateurs des années 50:

- Les éditeurs de liens (ou les lieurs)
- Les chargeurs
- Les contrôleurs

Ces outils étaient rudimentaires (pas trop d'options) et rigides (pas très interactifs)

- Dans les débuts des années 70 des outils plus conviviaux et plus fournis en options et facile à combiner dans un pipe ont vu le jour

- Les installateurs comme « grep » fournis avec Unix/Linux
- Les constructeurs tels que make



HISTOIRE :

FACTEURS ESSENTIELS DANS L'ÉVOLUTION

- L'évolution technologique offrant de meilleures possibilités d'ingénierie :
 - Des systèmes plus puissants en calcul et en stockage
 - Avènement du traitement parallèle
 - Partage et communication
- Meilleure compréhension du processus de développement :
 - Nouveaux paradigmes de programmations
 - Nouvelles méthodes de développement ...

Et depuis l'évolution continue ... les outils sont intégrés dans des environnements de développement « IDE »



DÉFINITION D'UN IDE

- En anglais «Integrated Development Environment »
 - Un system avec une interface utilisateur graphique permettant à un développeur d'effectuer l'ensemble des opérations liées à l'activité de développement de programmes ou application.
 - Il englobe au minimum les outils de base:
 - Edition du code source
 - Compilation
 - Teste et exécution
 - Débogage
 - Autres activités liées à la gestion du développement



L'ENVIRONNEMENT DE DÉVELOPPEMENT DU LOGICIEL

- Un logiciel est en pratique le résultat de l'exécution d'un code sur un ensemble de données en utilisant des dispositifs physiques adéquats
- Construire un système ne se réduit pas la simple écriture du code (programmation ou codage).
- Plusieurs facteurs interviennent dans la construction d'un logiciel
- Ces facteurs composent l'environnement de développement du logiciel



L'ENVIRONNEMENT DE DÉVELOPPEMENT DU LOGICIEL

- L'espace de travail «workspace»: espace de développeur crée le codage au jour le jour
- L'organisation: le développement s'effectue souvent dans le contexte d'une structure organisationnelle (équipes de développement logiciel, les essais en équipes, la commercialisation, suivi et support, etc.)
- L'Architecture: détermine la structure du produit final, ainsi que la structure du code et des différentes versions du produit final
- La maintenance du code (l'histoire du codage): Comprend les outils, les processus et les politiques de gestion des configurations



EXEMPLE D'INFLUENCE DE CES FACTEURS

- L'organisation et l'architecture influencent la façon de coder :
 - Une architecture moins étroitement couplé peut être développée en parallèle plus facilement qu'une architecture étroitement couplé.
 - Une petite équipe dans une chambre peut communiquer plus efficacement qu'une grande équipe éparpillée à travers le monde
- Dans chaque situation, la manière de gérer l'activité de développement nécessite une politique adéquate



L'ESPACE DE TRAVAIL

- Regroupe les outils permettant au minimum
 - L'édition et structuration du code : exemple
 - Emacs
 - La compilation et édition de liens: exemple
 - GNU toolchain (Make, automake, gcc)
 - Microsoft Visual Studio
 - CodeWarrior
 - Xcode
 - Le «debugging» : exemple
 - Gdb
 - GNU Binutils



L'ESPACE DE TRAVAIL

- En plus il doit intégrer un «kit de développement logiciel» (Software Development Kit)
- SDK est un ensemble de bibliothèques de fonctionnalité qui permettent à un développeur de coder des applications dans un langage donné pour
 - Des plateformes logicielles générales (orienté objet, fonctionnelle...etc)
 - Des plateformes logicielles particulières (framework COM, CORBA, XML based frameworks, web...)
 - Des plateformes matérielles particulières (Grille de calcul, réseaux mobile, console de jeu vidéo, le système embarqué)
- Exemples (un peu viyoytes)
 - Le SDK DirectX de Microsoft
 - Le SDK Java de Sun Microsystems
 - La Net Yaroze pour PlayStation de Sony (1997)

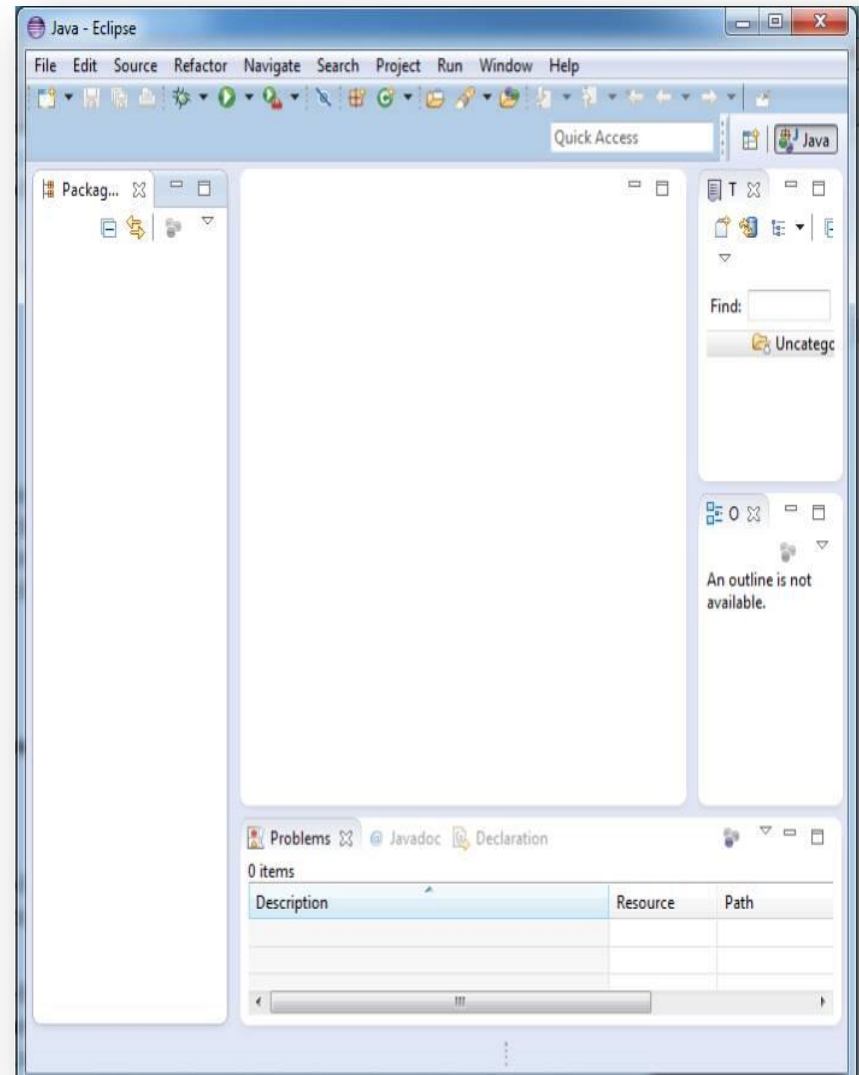


ESPACE DE TRAVAIL : EXEMPLES

Espace de travail MS V. Studio



Espace de travail Eclipse



L'ARCHITECTURE

- Selon le standard IEEE Std. 610.12, l'architecture est la structure organisationnelle d'un système de composants.
“The organizational structure of a system of component.”
- La plupart des définitions présentent l'architecture comme une vision des différents aspects de la structure du système
- L'architecture impose des contraintes sur la manière (où et comment) de monter un système à partir d'un ensemble de fonctionnalités (sous forme de briques) qui s'emboîtent en facilitant leur future maintenance (ajout, suppression, remplacement...)

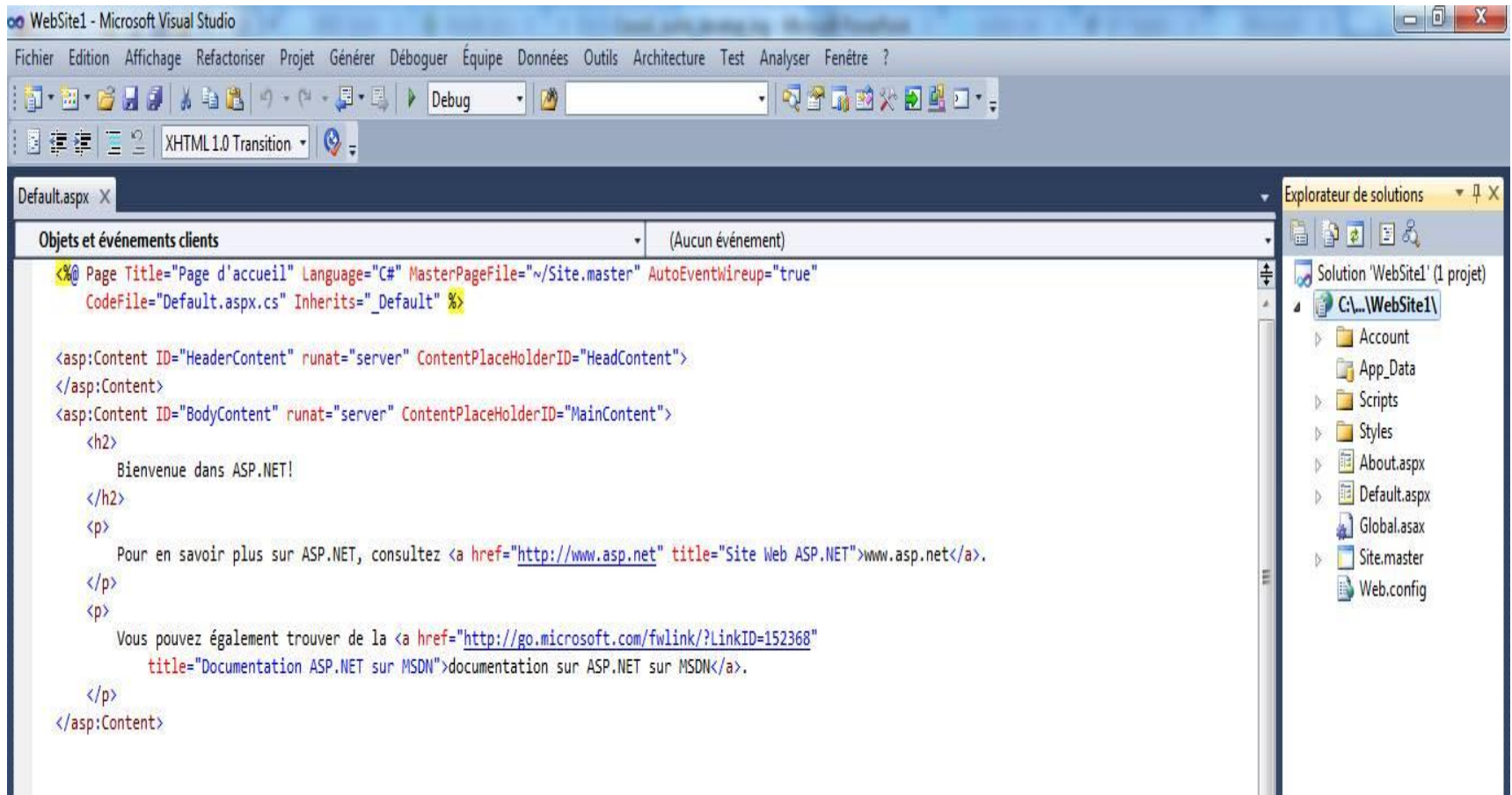


L'ARCHITECTURE DÉCRIT :

- Une décomposition du logiciel en différentes parties
- La façon dont ces parties s'emboîtent pour former le logiciel global
- Les directives sur la façon dont le système doit être modifiée.
- La façon d'organiser le développement :
 - Que doit contenir le livrable (produit final)
 - Les éléments de communication entre les parties du logiciel
 - Indirectement, la structure de répertoire et d'autres aspects structuraux liés à la gestion et la maintenance du code source. (En particulier les sauvegardes et l'historique)

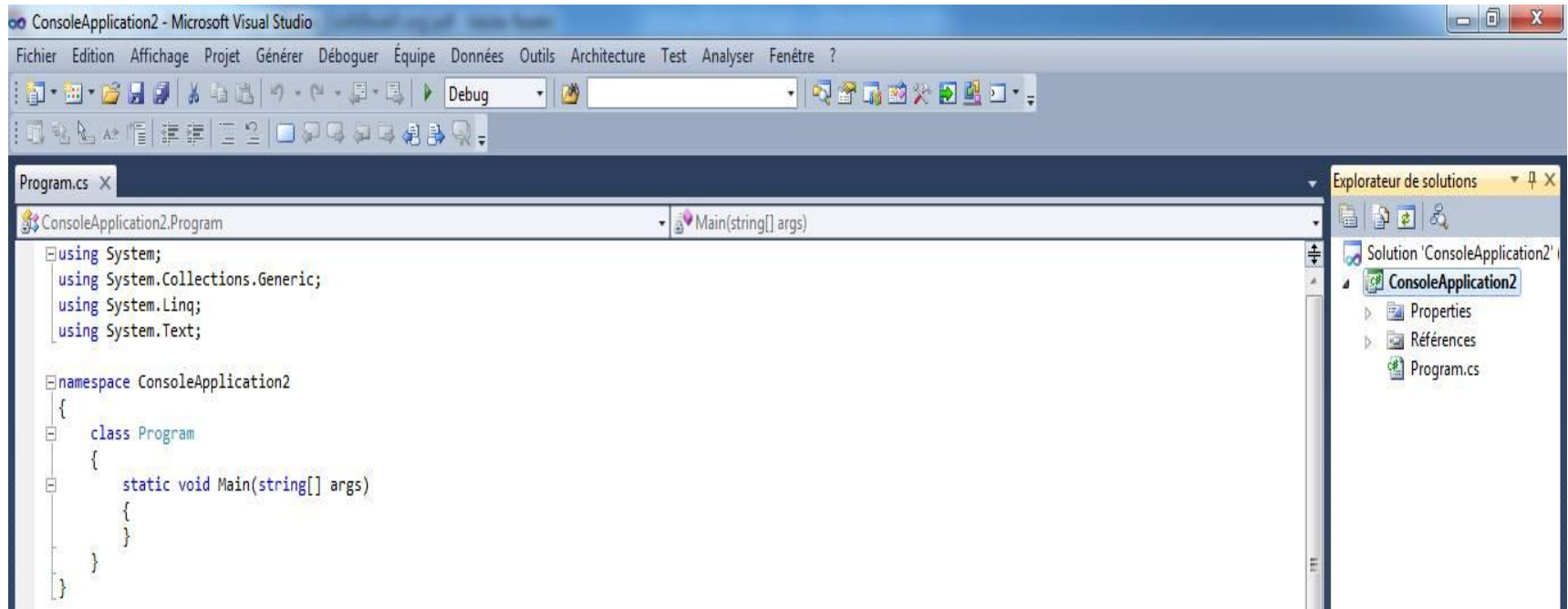


Environnement pour une application WEB



L'ARCHITECTURE : EXEMPLE

Environnement pour une application Console



DIFFÉRENTS OUTILS INTÉGRÉS À L'IDE

- Outil pour le développement :
 - Gestionnaire de l'interface utilisateur (UIMS)
 - Éditeur de code
 - Compilateurs
 - Éditeurs de lien
 - Préprocesseur
 - Debuggeur niveau code source
 - Debuggeur niveau exécutable



DIFFÉRENTS OUTILS INTÉGRÉS À L'IDE

- Outil d'ingénierie:
 - Constructeur (Builders)
 - Gestionnaires de version
 - Assistant de conception (outil CASE)
 - Générateurs de code
 - Générateurs de Tests
 - Outils d'analyse (retro ingénierie)



DIFFÉRENTS OUTILS INTÉGRÉS À L'IDE

- Outil de gestion :
 - Gestionnaire de tâches (liste à faire)
 - Outils de prise en charge du processus de développement (support de langage de modélisation de processus : BPMN)
 - Outils pour les diagrammes de Gantt et de PERT
 - Outils d'estimation des coûts sur la base de modèles, tels que COCOMO (COConstructive COst Model)
 - Estimer l'effort et la durée de développement logiciel selon les ressources allouées (Barry Boehm)

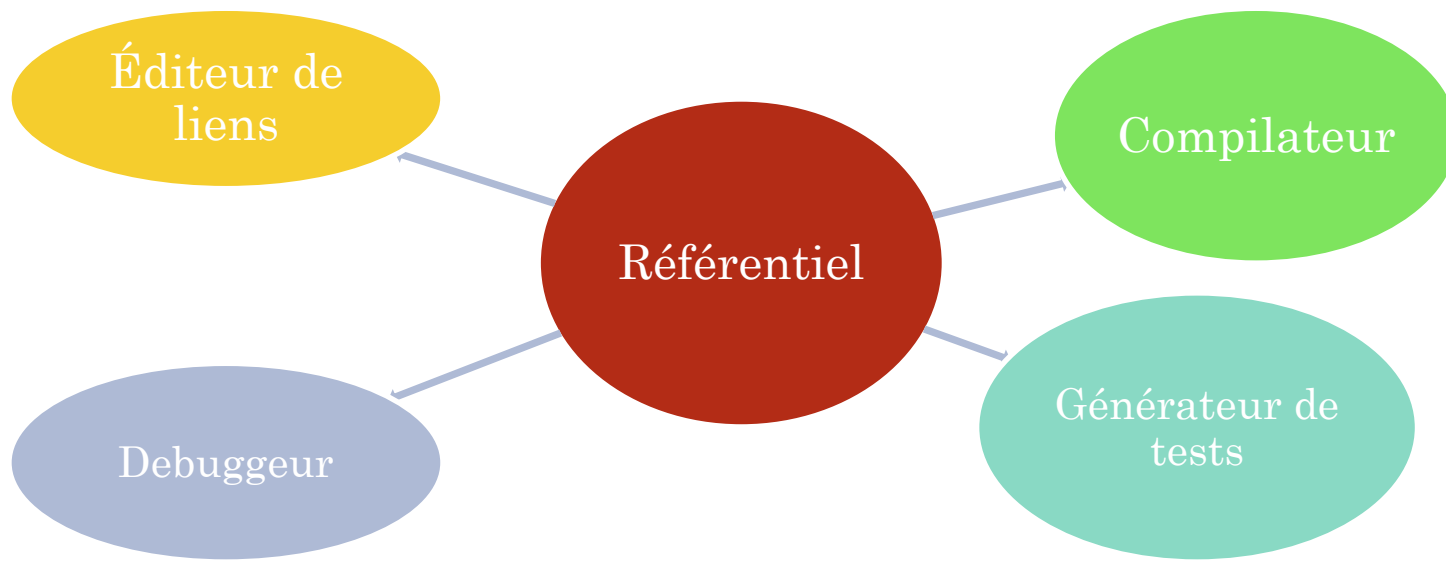


APPROCHES D'INTÉGRATION DES OUTILS

- Par Échange de fichiers texte:

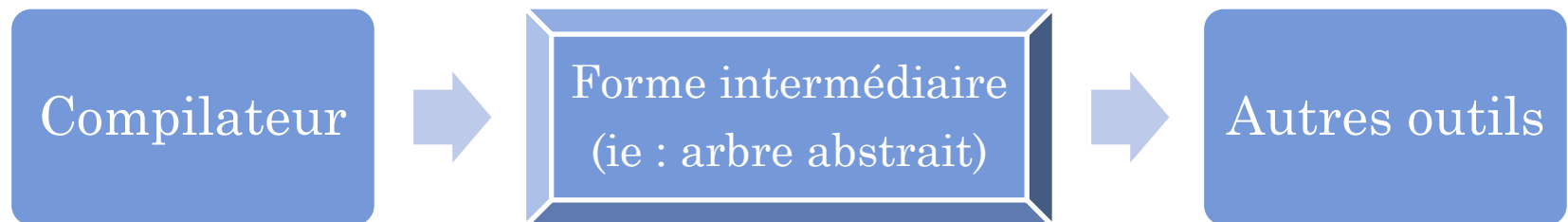


- Référentiel d'échange de données entre divers outils

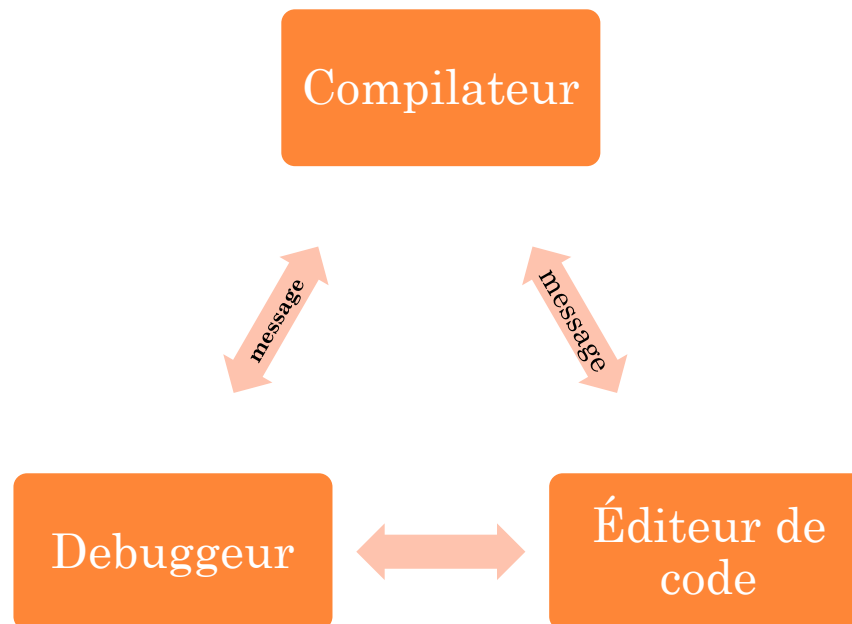


APPROCHES D'INTÉGRATION DES OUTILS

- Sorties et interfaces standard



- Échange de messages



CRITÈRES DE CHOIX DES IDE

- Le mode d'interaction:
 - outils orientés lots de commande (batch ou script)
 - outils interactifs (avec GUI)
- Niveau de formalité:
 - syntaxe / sémantique des documents produits
- Degré de couplage avec les étapes du cycle de vie
- Degré de standardisation
- Mode opératoire:
 - Statique versus dynamique
- Catégorie d'outils intégrés :
 - Outil de développement
 - Outil de production final
 - Outil de gestion ...
- Mode d'utilisation:
 - Mono utilisateur
 - Multi utilisateurs
- Mode d'installation :
 - Mono poste
 - Réseau

