

Technologies Web synchrones et multi-dispositifs

Introduction au cours

<https://aurelient.github.io/react-iga/2019/>

Plan

- ▶ Introduction au cours
- ▶ L'informatique Ubiquitaire
- ▶ Rappels Stack Javascript

Technologies Web synchrones et multi-dispositifs

Objectifs:

- ▶ Aller au delà de “*l'état de l'art*”. Concevoir et réaliser des applications
- ▶ Réfléchir aux enjeux de l'informatique ambiante

Stack d'outils de programmation pour JS mobile et embarqué, temps réel, multi-dispositif et collaboratif.

Contenu de l'UE

Outils de programmation en JS

Programmation réactive (React et Redux)

Multi-dispositif

Nouvelles modalités de présentation et d'interaction

TP : Problèmes de rendu à éviter

- ▶ Impossible de cloner le projet
 - ▶ Erreur dans l'identifiant du projet
 - ▶ Nous n'avons pas les droits d'accès au projet
- ▶ Manque de doc
 - ▶ Placez tous les fichiers dans le repo du projet
 - ▶ README.md obligatoire et clair sur les étapes à suivre pour faire tourner votre code
- ▶ Notre environnement est différent du votre
 - ▶ URLs absolues dans le code
 - ▶ Ne mettez pas le contexte en dur dans votre code
 - ▶ Déployez votre projet sur une autre machine, et testez toutes les fonctionnalités.
 - ▶ Testez sur plusieurs navigateurs
 - ▶ "Ça marche chez moi" n'est pas un bon argument

Plan

- ▶ Introduction au cours
- ▶ **L'informatique Ubiquitaire**
- ▶ Rappels Stack Javascript

L'informatique Ubiquitaire

Un peu d'histoire:

- ▶ Mark Weiser et le Xerox PARC
- ▶ En Europe :
 - ▶ i-LAND
 - ▶ Phillips
- ▶ Aujourd'hui

L'Ubicomp : 3^e ère de l'informatique

Fin des années 1980 - début 90 :

- ▶ L'informatique personnelle s'impose (Mac: 84, Windows 3: 90)
- ▶ Miniaturisation de l'informatique
- ▶ Informatique embarquée, premier téléphone mobile, Palm...
- ▶ Développement d'interfaces utilisateurs grand public
- ▶ Salles interactives, réalité augmentée...

De nombreux termes:

- ▶ Ubiquitous computing (Xerox)
- ▶ Pervasive computing (IBM)
- ▶ Intelligence ambiante (EU)

Mark Weiser – Le père de l'Ubicomp

The Computer for the 21st Century

Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence

by Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

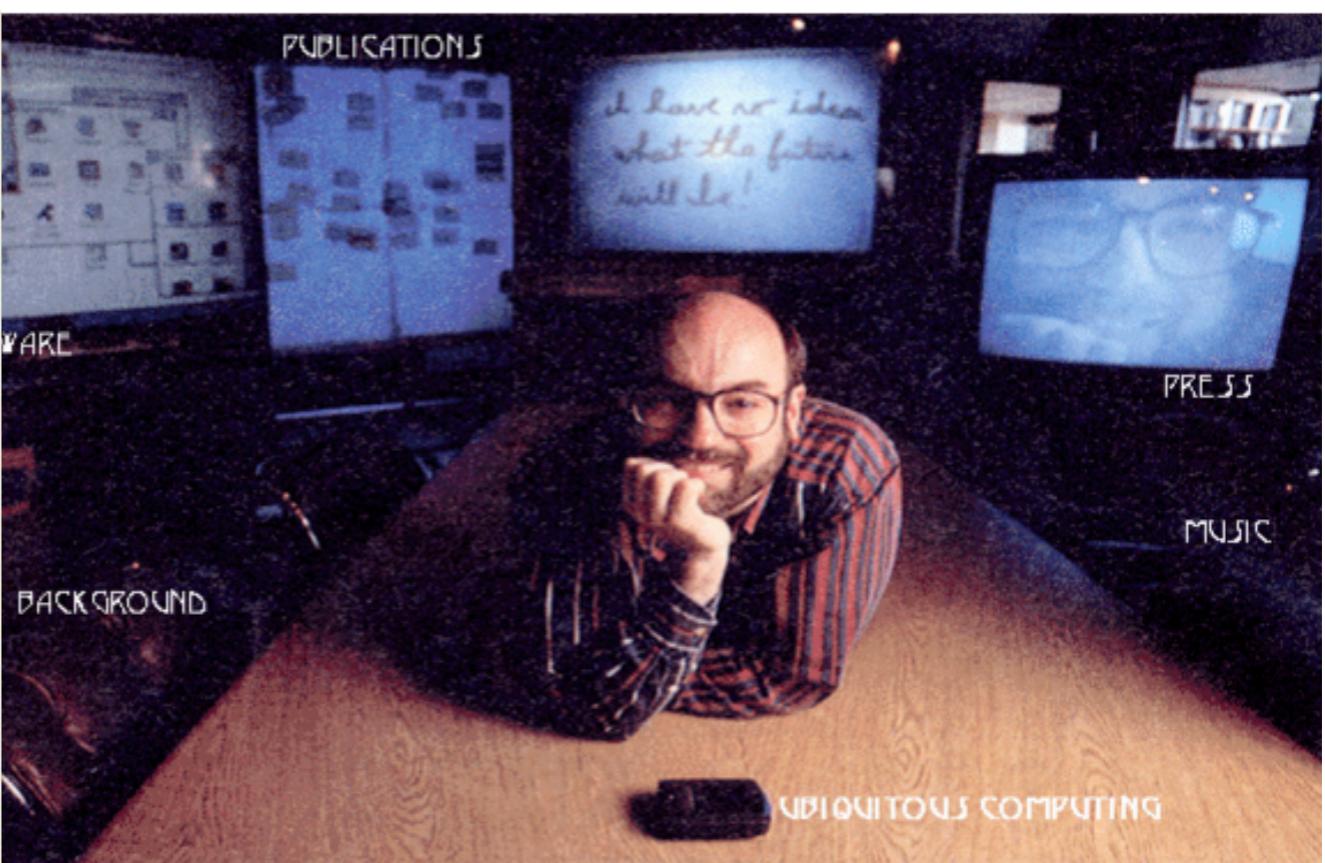
Consider writing, perhaps the first information technology. The ability to represent spoken language symbolically for long-term storage freed information from the limits of individual memory. Today this technology is ubiquitous in industrialized countries. Not only do books, magazines and newspapers convey written information, but so do street signs, billboards, shop signs and even graffiti. Candy wrappers are covered in writing. The constant background presence of these products of "literacy technology" does not require active attention, but the information to be transmitted is ready for use at a glance. It is difficult to imagine modern life otherwise.

Silicon-based information technology, in contrast, is far from having become part of the environment. More than 50 million personal computers have been sold, and the computer nonetheless remains largely in a world of its own. It is approachable only through complex jargon that has nothing to do with the tasks for which people use computers. The state of the art is perhaps analogous to the period when scribes had to know as much about making ink or baking clay as they did about writing.

The arcane aura that surrounds personal computers is not just a "user interface" problem. My colleagues and I at the Xerox Palo Alto Research Center think that the idea of a "personal" computer itself is misplaced, and that the vision of laptop machines, dynabooks and "knowledge navigators" is only a transitional step toward achieving the real potential of information technology. Such machines cannot truly make computing an integral, invisible part of people's lives. We are therefore trying to conceive a new way of thinking about computers, one that takes into account the human world and allows the computers themselves to vanish into the background.

Such a disappearance is a fundamental consequence not of technology but of human psychology. Whenever people learn something sufficiently well, they cease to be aware of it. When you look at a street sign, for example, you absorb its information without consciously performing the act of reading. Computer scientist, economist and Nobelist Herbert A. Simon calls this phenomenon "compiling"; philosopher Michael Polanyi calls it the "tacit dimension"; psychologist J. J. Gibson calls it "visual invariants"; philosophers Hans-Georg Gadamer and Martin Heidegger call it the "horizon" and the "ready-to-hand"; John Seely Brown of PARC calls it the "periphery." All say, in essence, that only when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals.

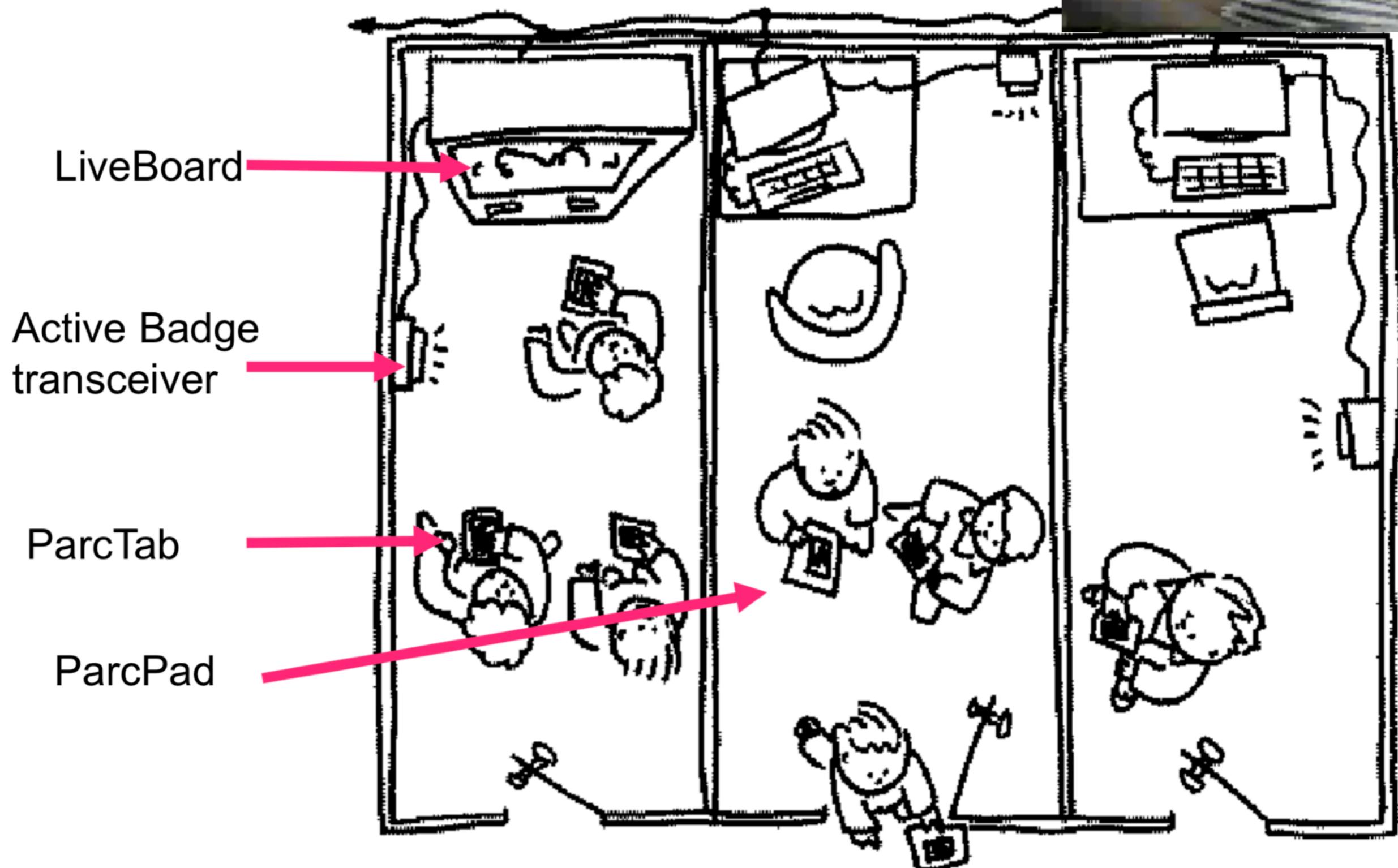
MARK WEISER is head of the Computer Science Laboratory at the Xerox Palo Alto Research Center. He is working on the next revolution of computing after workstations, variously known as ubiquitous computing or embodied virtuality. Before working at PARC, he was a professor of computer science at the University of Maryland; he received his PhD from the University of Michigan in 1979. Weiser also helped found an electronic publishing company and a video arts company and claims to enjoy computer programming "for the fun of it." His most recent technical work involved the implementation of new theories of automatic computer memory reclamation, known in the field as garbage collection.



Formule une vision de l'informatique diffuse, dans l'environnement et les objets de tous les jours.

<https://www.lri.fr/~mbl/Stanford/CS477/papers/Weiser-SciAm.pdf>

À Xerox Parc



Computing by the inch, foot, & yard

Ubiquitous computing @ Xerox PARC, 1988 - 1995

Devices according to model size approach:

PARCTab



Inch-sized

PARCpad



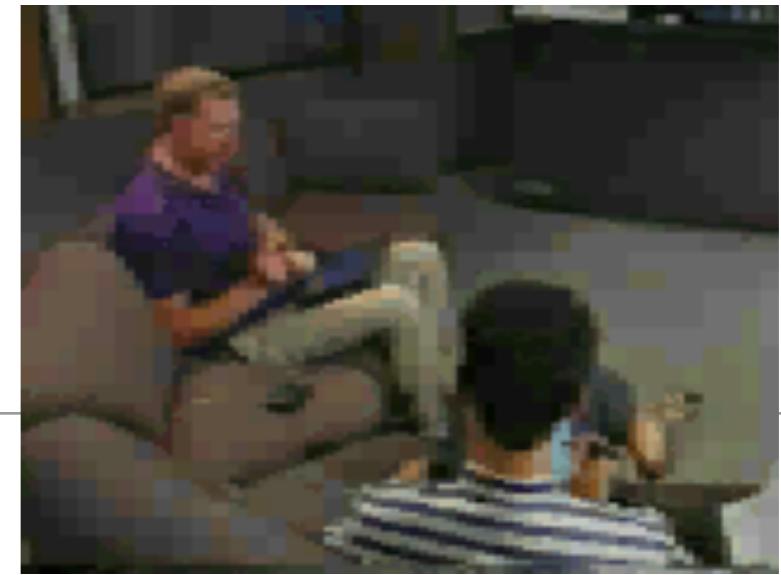
Foot-sized

Liveboard



Yard-sized

Échelle, continuité



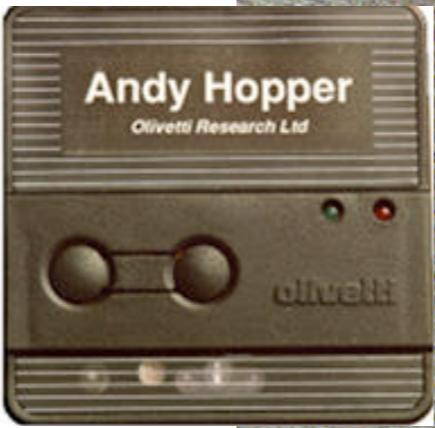
tiny Every pixel counts Context Aware Pervasive	~ book size Action at point of input Error handling & disambiguation Disposable or personal	> meter Fluid interaction Freeform input Shared collaborative
PARC Tabs <small>hundreds/person</small>	PARC Pads <small>tens/person</small>	Liveboard <small>one/person</small>

Tabs

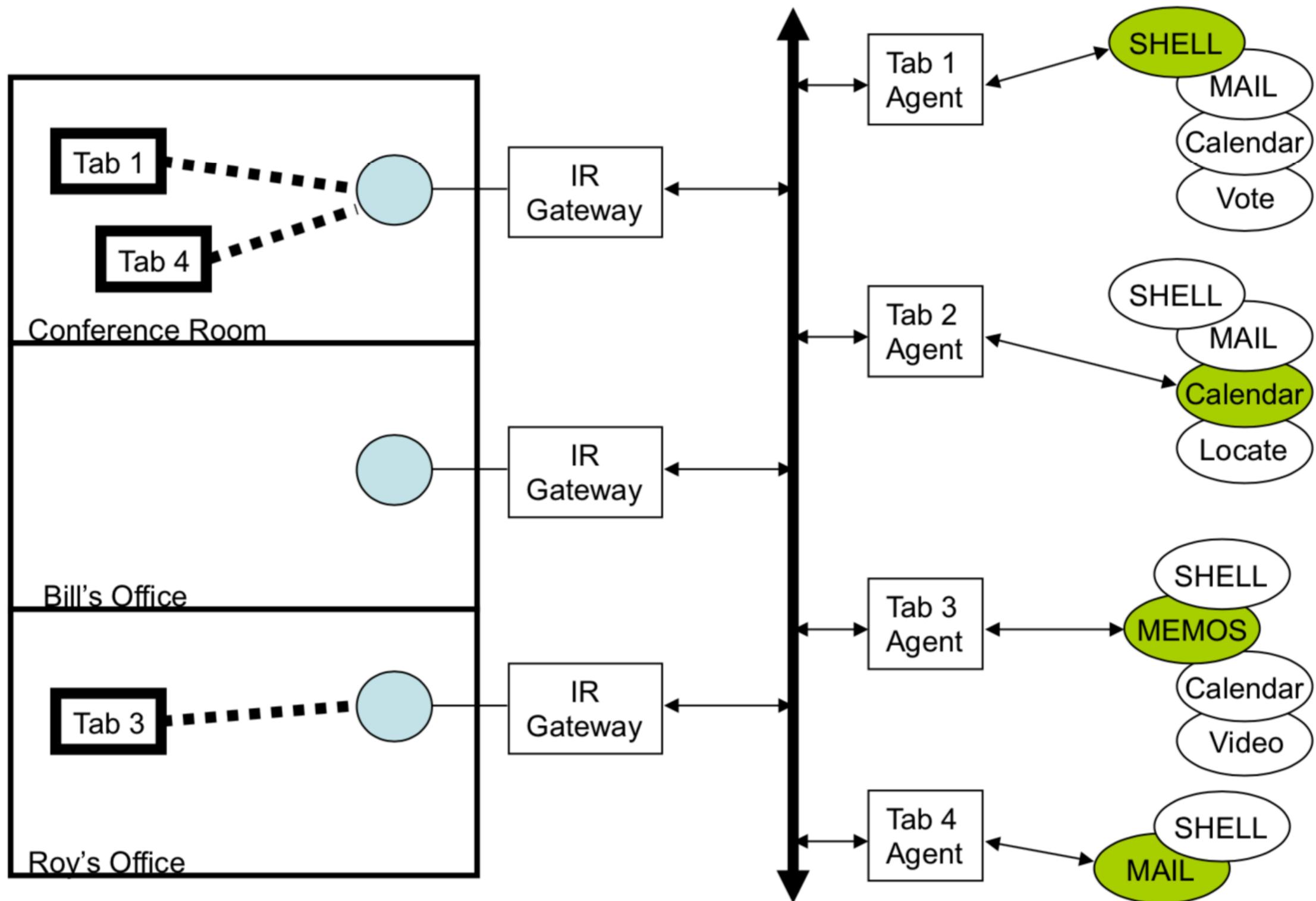
"Tabs are the smallest components of embodied virtuality. Because they are interconnected, tabs will expand on the usefulness of existing inch-scale computers such as the pocket calculator and the pocket organizer. Tabs will also take on functions that no computer performs today."

M. Weiser

- :: Smallest components
- :: Interconnected
- :: Embedded applications



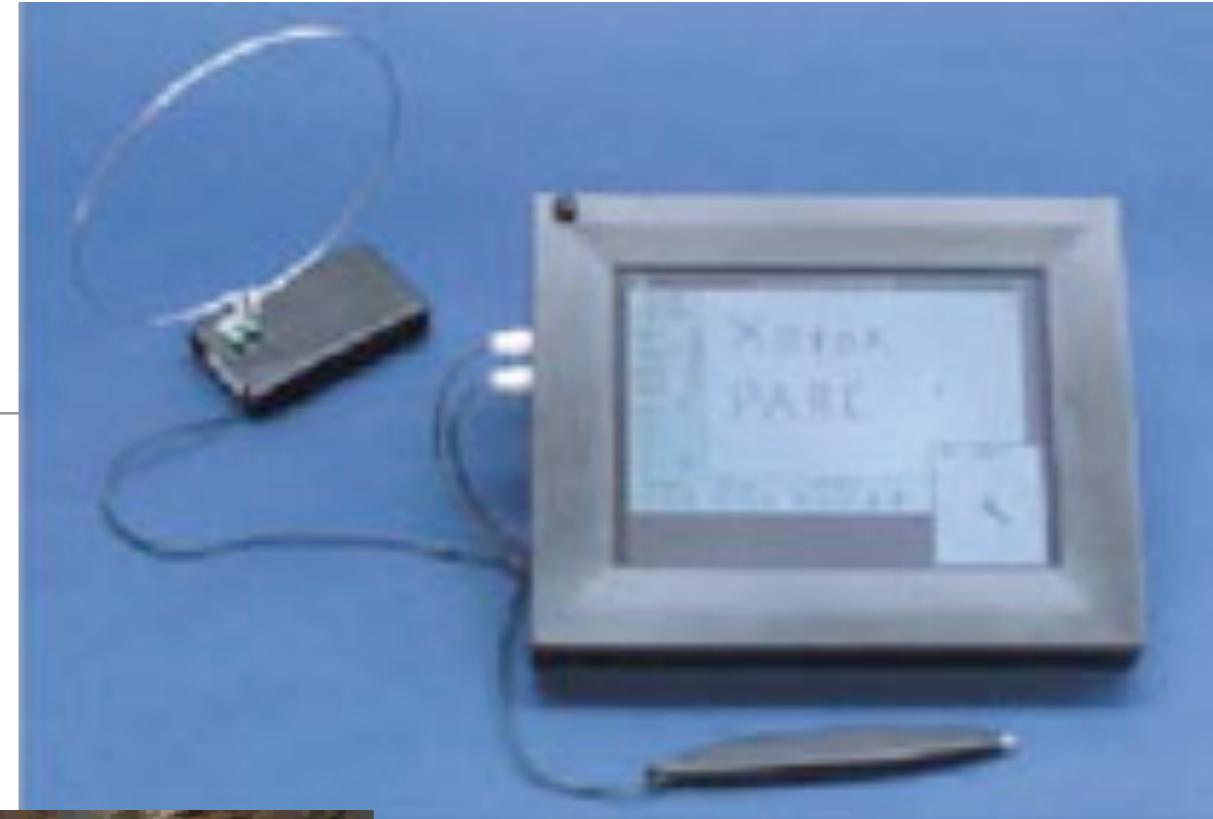
ParcTab Transceiver IR Gateway Ethernet Agent Applications



Pads

"Pads differ from conventional portable computers in one crucial way. Whereas portable computers go everywhere with their owners, the pad that must be carried from place to place is a failure. Pads are intended to be 'scrap computers' (analogous to scrap paper) that can be grabbed and used any-where; they have no individualized identity or importance."

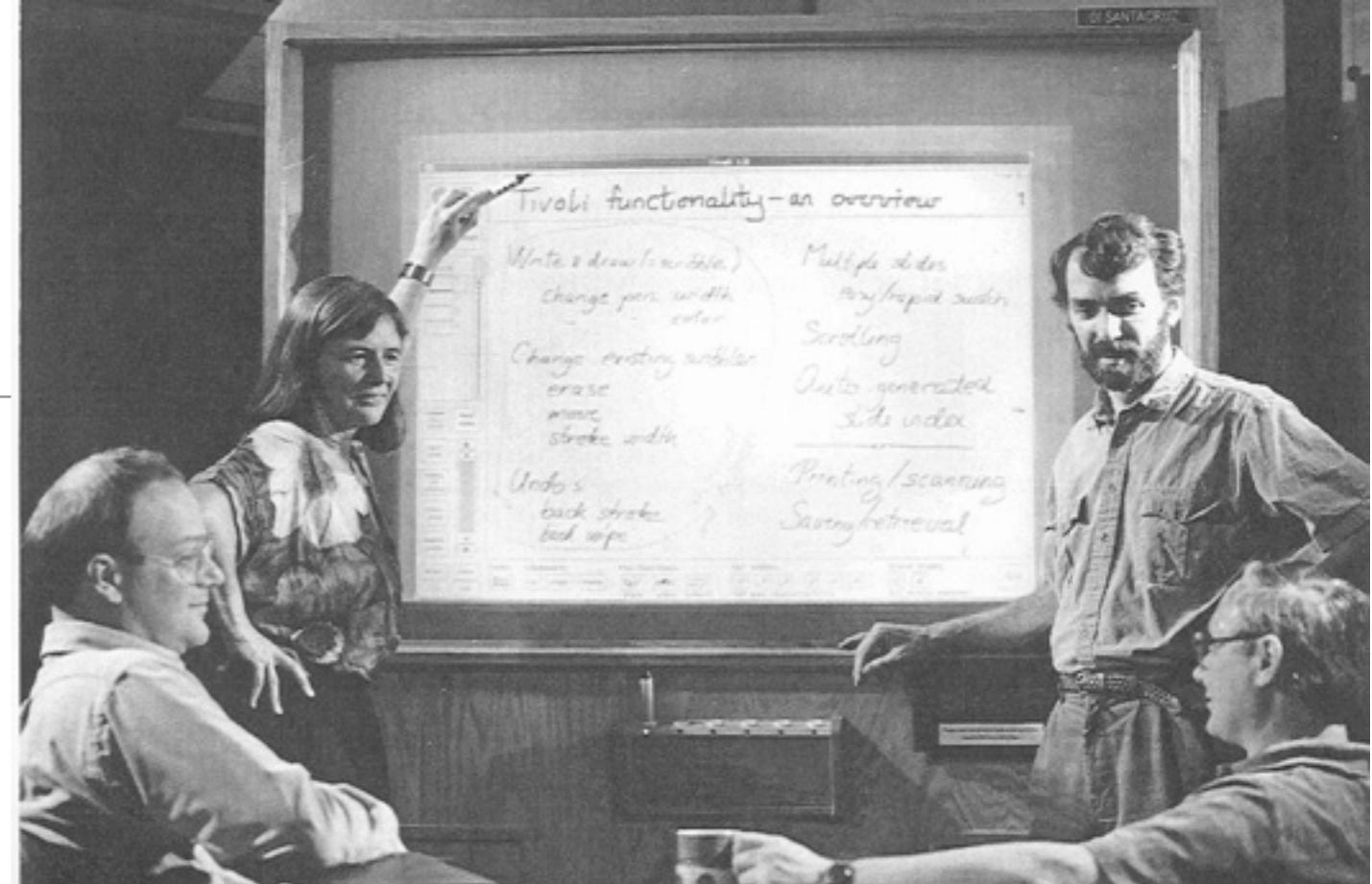
M. Weiser



Boards

"We have built enough Liveboards to permit casual use: they have been placed in ordinary conference rooms and open areas, and no one need sign up or give advance notice before using them. By building and using these boards, researchers start to experience and so understand a world in which computer interaction casually enhances every room."

M. Weiser



Xerox's Liveboard with Tivoli application



The Computer for the 21st Century

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”

- ▶ l'écriture, l'eau, l'électricité...
- ▶ Embarqué, invisible, tacite, ambiant, périphérique.
- ▶ Context aware

Dangling Wire de Natalie Jeremijenko



Un exemple de Calm Technology

<https://people.csail.mit.edu/rudolph/Teaching/weiser.pdf>
<https://calmtech.com/index.html>

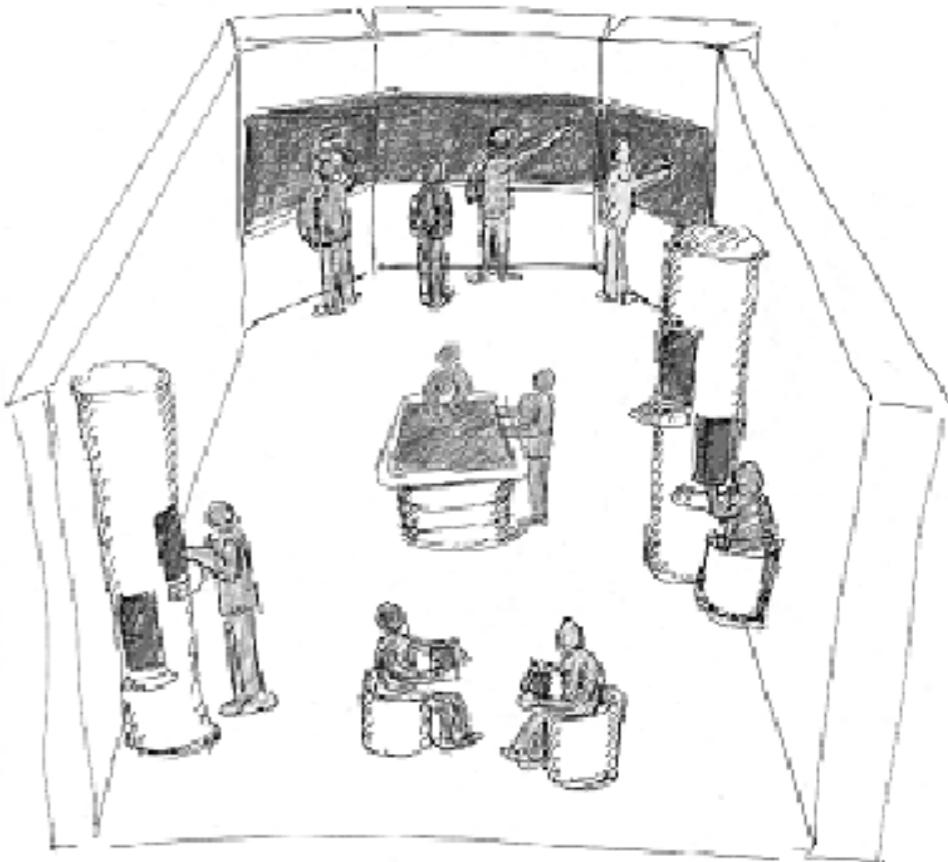
L'informatique Ubiquitaire

Un peu d'histoire:

- ▶ Mark Weiser et le Xerox PARC
- ▶ En Europe :
 - ▶ i-LAND
 - ▶ Phillips
- ▶ Aujourd'hui

En Europe

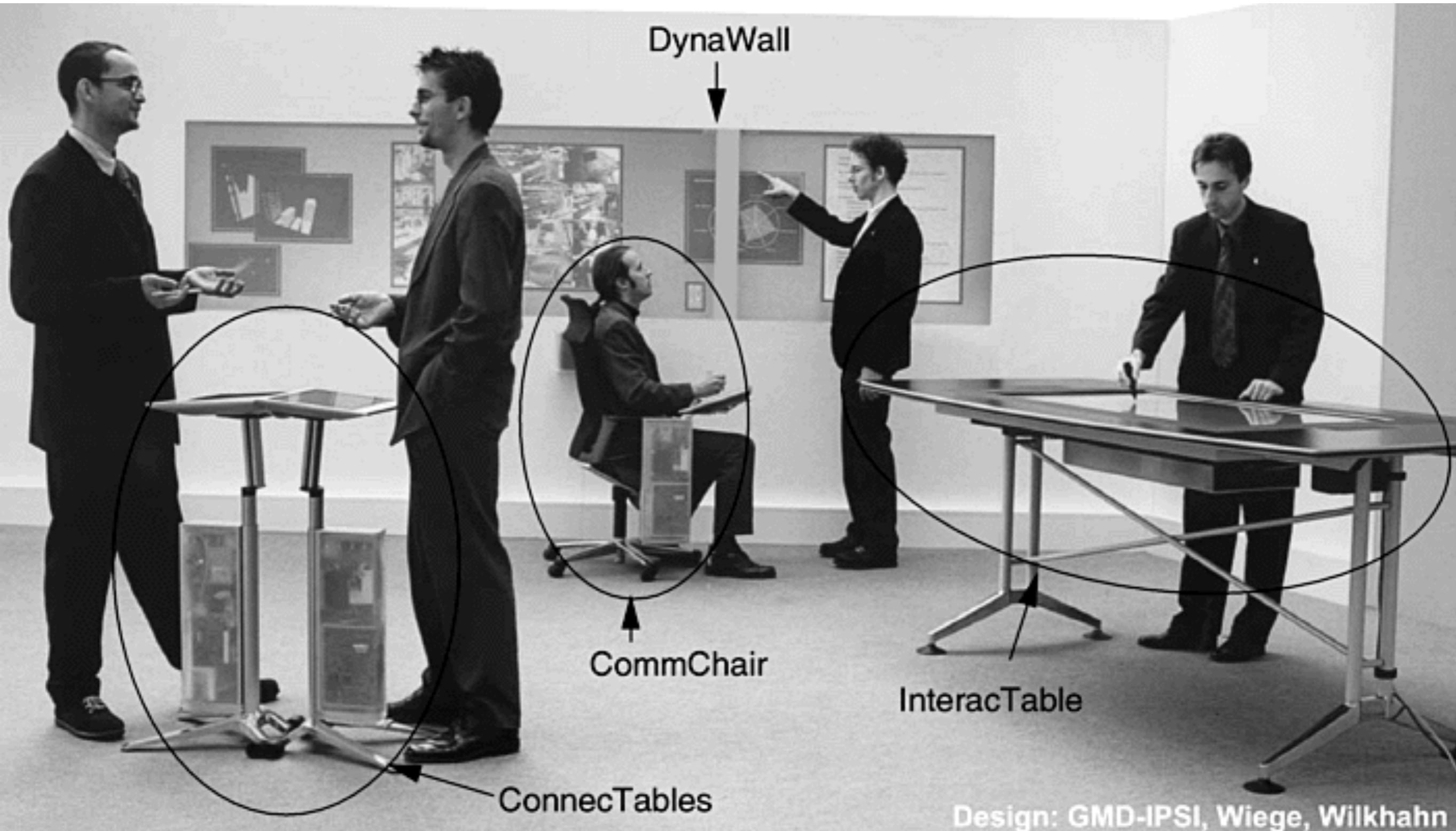
i-Land: “RoomWare” pour la créativité et les Concepts



- ▶ Intégration de l'espace architectural et informationnel.
- ▶ Nouvelles pratiques de travail.

Scénarios

- ▶ Rencontre dans le couloir, dessin sur le mur
- ▶ Travail collaboratif en sous-groupe → session de discussion



i-Land

DynaWall

- ▶ Mur interactif tactile : 4,5m x 1,1m

CommChair

- ▶ Chaises avec ordinateur et station de docking intégrées pour portables

InteracTable

- ▶ Table interactive tactile : 65x85cm

Passage

- ▶ Mécanisme de passage d'information et d'association entre objets physiques et numériques.

Infrastructure

Integration

- ▶ De tous les composants hardware
- ▶ Réseau + infrastructure logicielle
- ▶ “OS for RoomWare”

Software Infrastructure ~ BEACH

- ▶ Partage d'information
- ▶ Gestion d'interfaces distribuées
- ▶ Distribution, réplication, gestion d'objet informationels

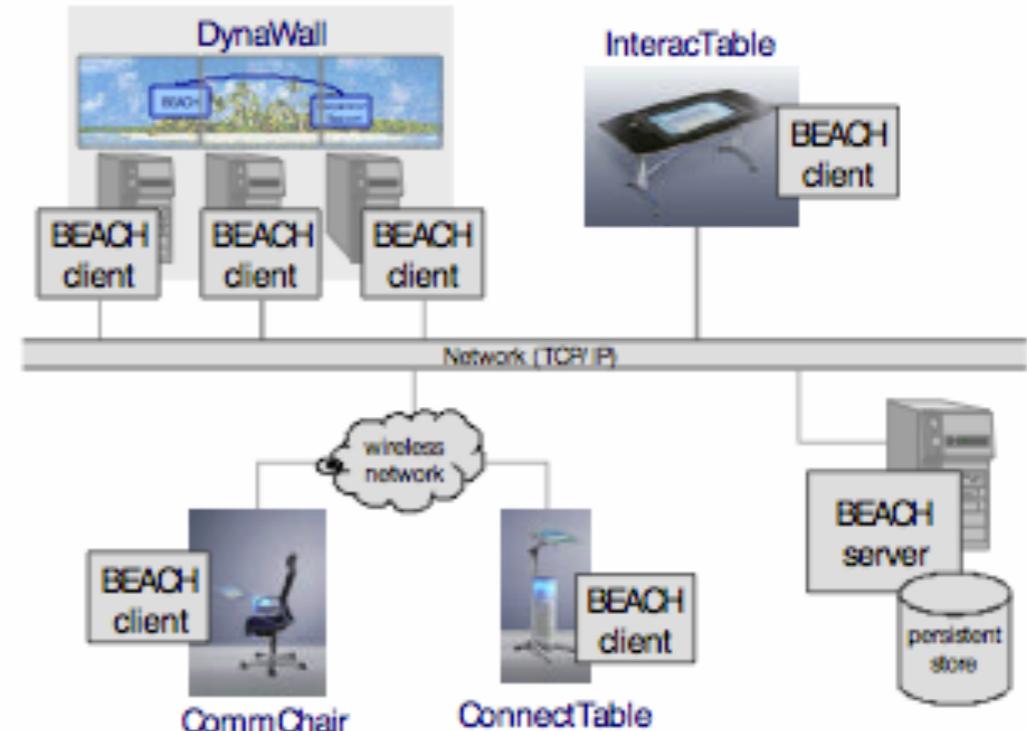


Fig. 4. BEACH clients running on different roomware component are synchronised by a server

Vision of the future (Philips, 1996)



VISION OF THE FUTURE

L'informatique Ubiquitaire

Un peu d'histoire:

- ▶ Mark Weiser et le Xerox PARC
- ▶ En Europe :
 - ▶ i-LAND
 - ▶ Phillips
- ▶ Aujourd'hui

Principes et exemples

Décentralisation

- ▶ Nombreux dispositifs
- ▶ Distribués, mobiles, p2p

Diversifiés

- ▶ Universel -> dédié
- ▶ Grand nombre de clients

Connectivité

- ▶ Toujours actif
- ▶ Sans fil

Simple

- ▶ “Information appliances”, dédiées avec fonctionnalité limitées
- ▶ “Intelligent”
 - Context-awareness
 - Activity-awareness

Une personne, de nombreux dispositifs



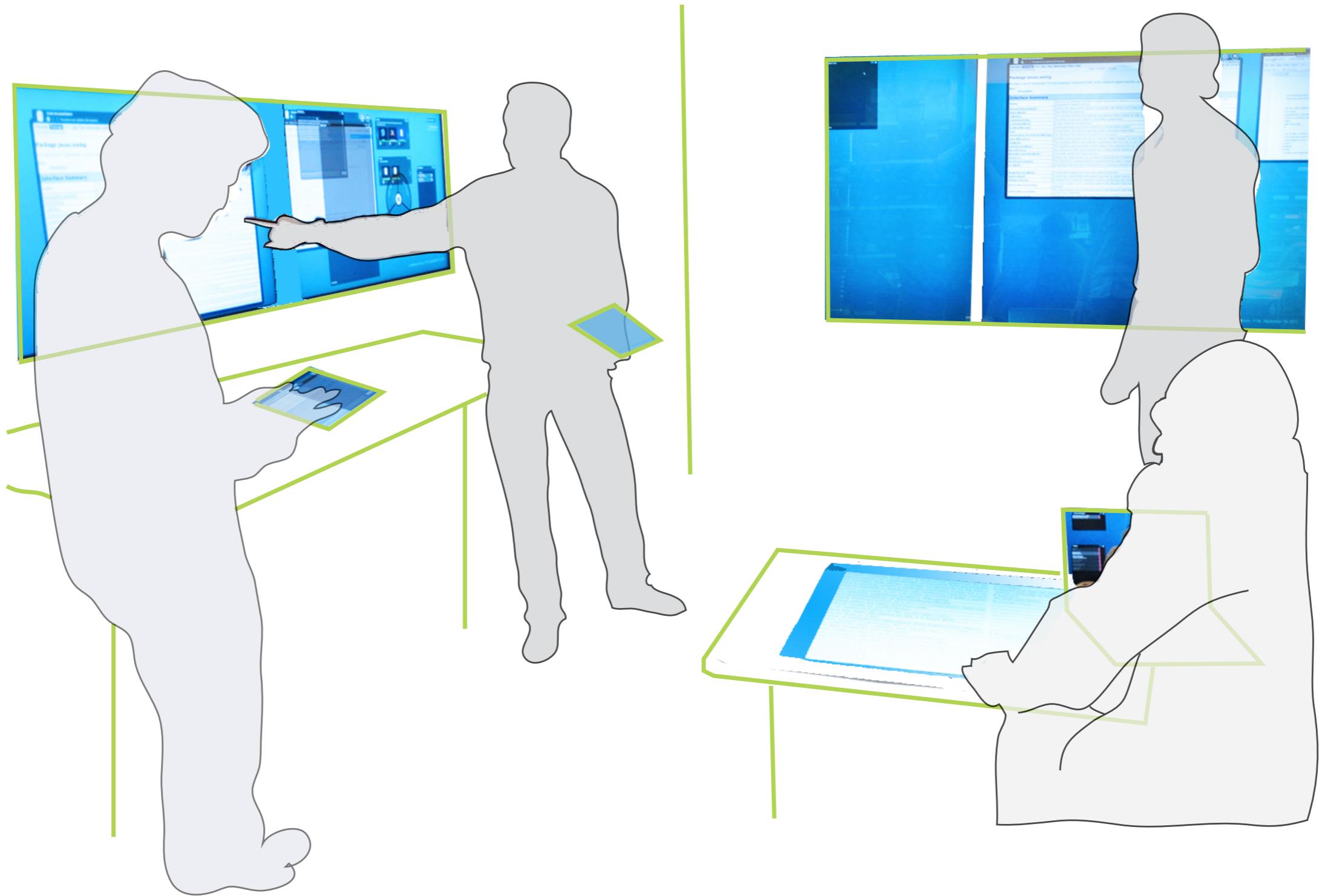
Un dispositif, de nombreux utilisateurs



Smart-home



Smart spaces



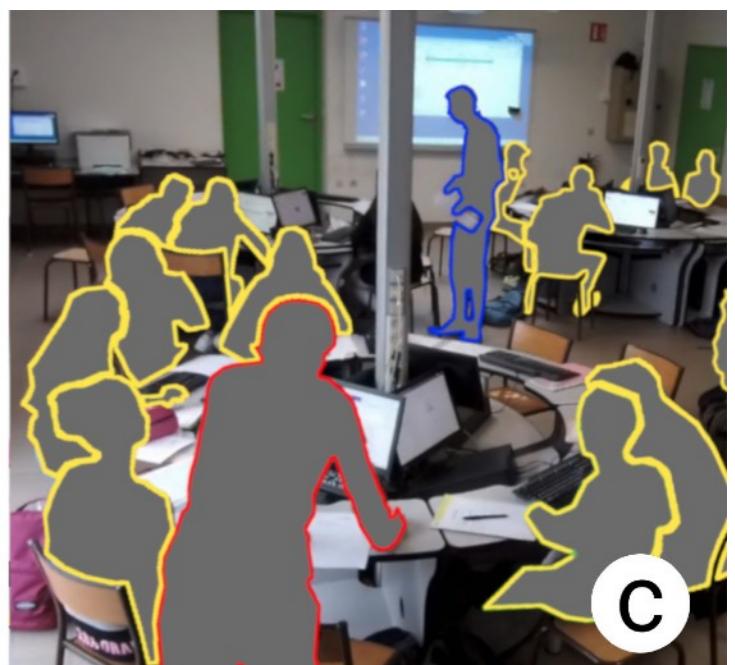
Mobilité



Santé et soins



Education



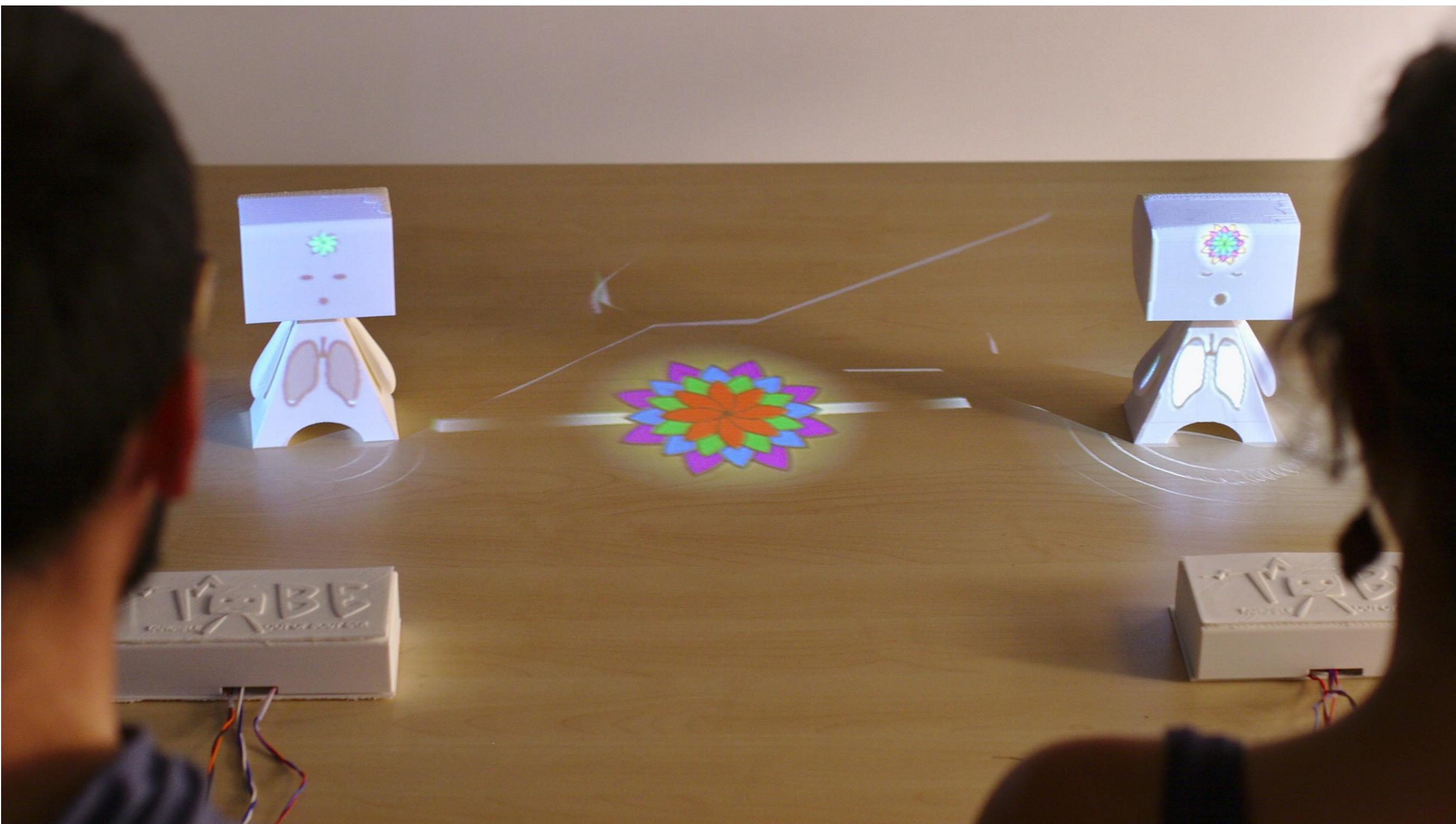
Wearables



Informatique tangible

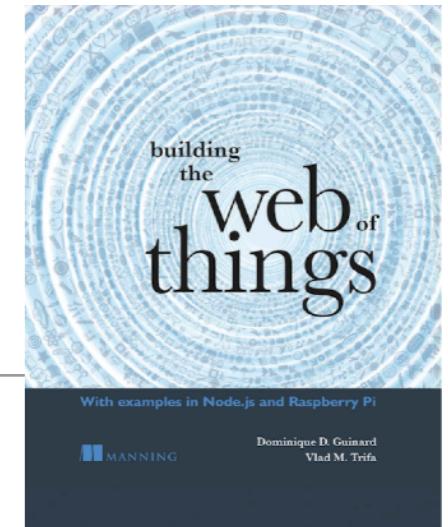


Réalité mixte



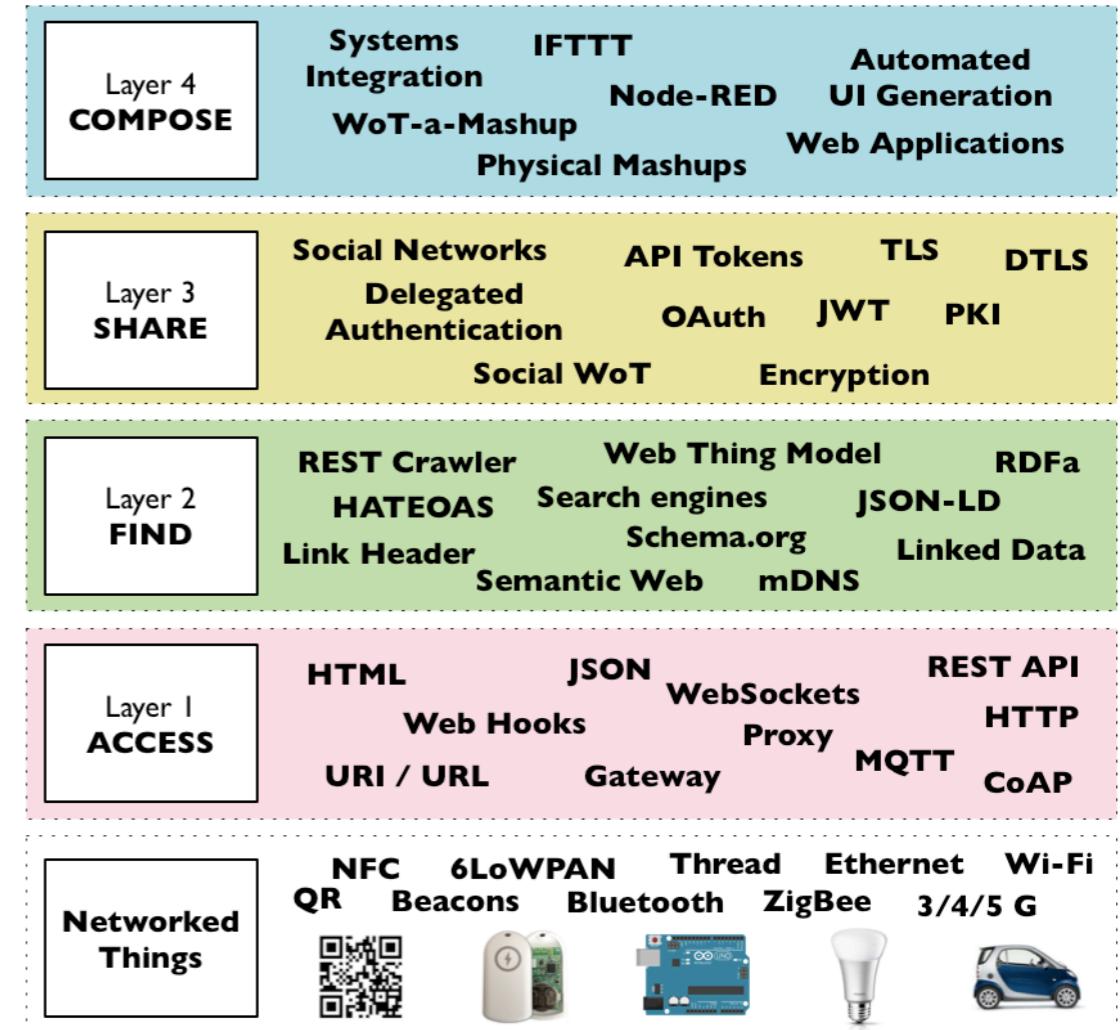
Web of Things

<https://www.w3.org/WoT/>



Couche Web sur l'Internet of Things

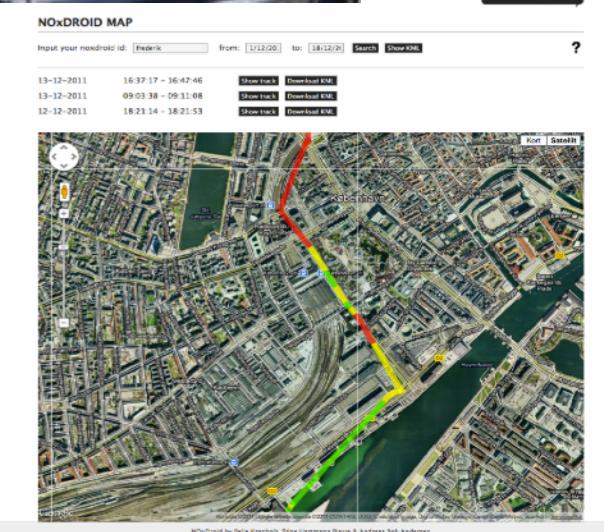
- ▶ Un GT du W3C
- 2 Candidate Recommendations
- ▶ Réutiliser les standards
- ▶ Simplifier la création de services
- ▶ Des bases anciennes
 - Projet Cooltown de HP en 2002
- ▶ Des entreprises actives:
 - xively, evrythng, ...



Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

Green Ubicomp

- ▶ Suivi de la consommation d'énergie (capteurs)
 - ▶ Chauffage / refroidissement intelligent
- ▶ Suivi de la pollution
- ▶ Gestion des déchets
- ▶ Suivi environnemental
- ▶ Visualisation -> motivation
 - > changement de pratiques
 - > choix politiques



Bilan

1. Concevoir pour la présence quasi-continue du numérique
2. Présenter l'information en respectant l'attention des utilisateurs -> de la périphérie vers le focus
3. Lier mondes physique et numérique
4. Transformer nos méthodes de conception, pour imaginer répondre à des besoins diffus et difficilement formalisables
5. Enjeux en termes de vie privée et environnemental

Plan

- ▶ Introduction au cours
- ▶ L'informatique Ubiquitaire
- ▶ **Rappels Stack Javascript du cours**

Node.js



Plateforme d'exécution basée sur le moteur Chrome

Permet de :

- ▶ mettre en place de “grosses” applications en **JS** et **modulaires**
- ▶ programmer dans le **même langage** côté serveur et côté client

Npm



Gestionnaire de paquets pour node

S'utilise “à la” apt-get

- ▶ localement : npm install <package>
- ▶ globalement : npm install -g <package>

Description de l'application dans un fichier package.json

Express

Framework Web pour node.js peu contraignant.

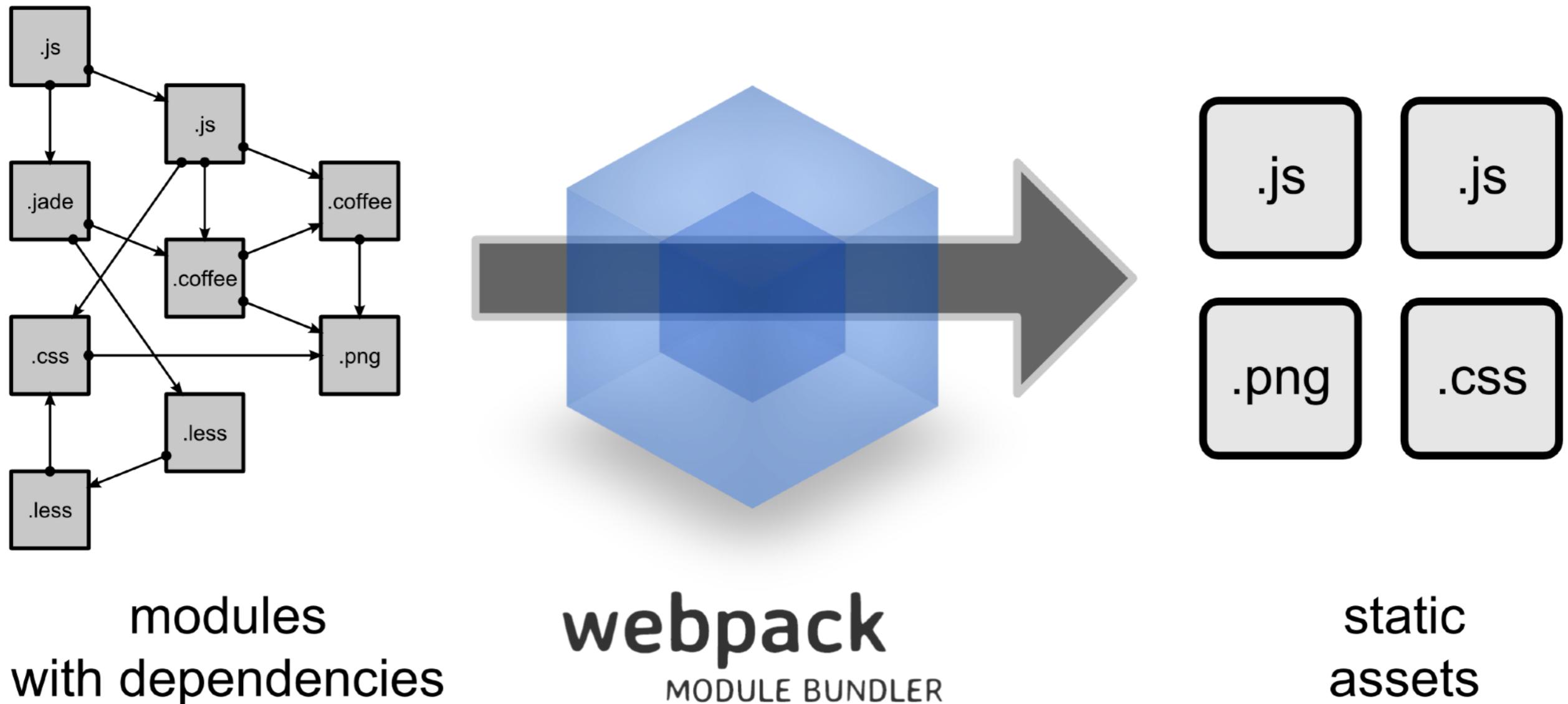
- ▶ Routage
- ▶ Rapide
- ▶ Négociation de contenu
- ▶ HTTP helpers (redirection, cache, etc)
- ▶ Compatible avec de nombreuses bibliothèques de templating

React

Framework JS

- ▶ DOM Virtuel
- ▶ Basés sur des composants
- ▶ Gestion des Vues + routeur + états
- ▶ ReactNative pour le mobile
- ▶ Redux pour la gestion avancée des états

Webpack



Webpack

- ▶ Permet la gestion de module (require / import)
- ▶ Découpe le code pour optimiser le chargement
- ▶ Injection de dépendance et multi-compilation
- ▶ S'intègre aux autres outils de build (Grunt/Gulp/etc)

Webpack

Write your code

app.js

```
import bar from './bar';  
  
bar();
```

bar.js

```
export default function bar() {  
  //  
}
```

Bundle with webpack

webpack.config.js

```
module.exports = {  
  entry: './app.js',  
  output: {  
    filename: 'bundle.js'  
  }  
};
```

page.html

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
    <script src="bundle.js"></script>  
  </body>  
</html>
```

Then run `webpack` on the command-line to create `bundle.js`.

Webpack et debug

Problème : tout est optimisé / minifié

Solution : création d'une source-map liant code bundlé au code "réel"

Des outils de dev :

<https://reactjs.org/blog/2019/08/15/new-react-devtools.html>

Autres outils JS

Tests : Mocha ou Jasmine

Vérification de code : ESLint

Compilation ES6 → ES5 : Babel

Gestion des styles : SASS