# Diseño e Implementación de un *HotSpot-In-a-Box*

Pau Oliva Fora

pau@eslack.org

4 de Febrero de 2004

# Introducción

1. **Objetivos**
2. **Conceptos**
3. **Diagrama de bloques**
4. **Hardware Wi-Fi soportado**
5. **Interfaz de configuración**
6. **Modos de funcionamiento**

# Objetivos

- Implementar un *gateway* o *access point controller* utilizando software libre
- Conseguir calidad técnica comparable a las soluciones comerciales existentes
- Desarrollo basado en la integración de componentes
- Realizar la función de punto de acceso añadiendo una tarjeta wi-fi sobre el mismo hardware
- Minimizar en la medida de lo posible la dependencia de un hardware determinado
- Proporcionar un método de instalación fácil y rápido
- Interfaz de configuración única y centralizada
- Obtener un software fácilmente personalizable y adaptable

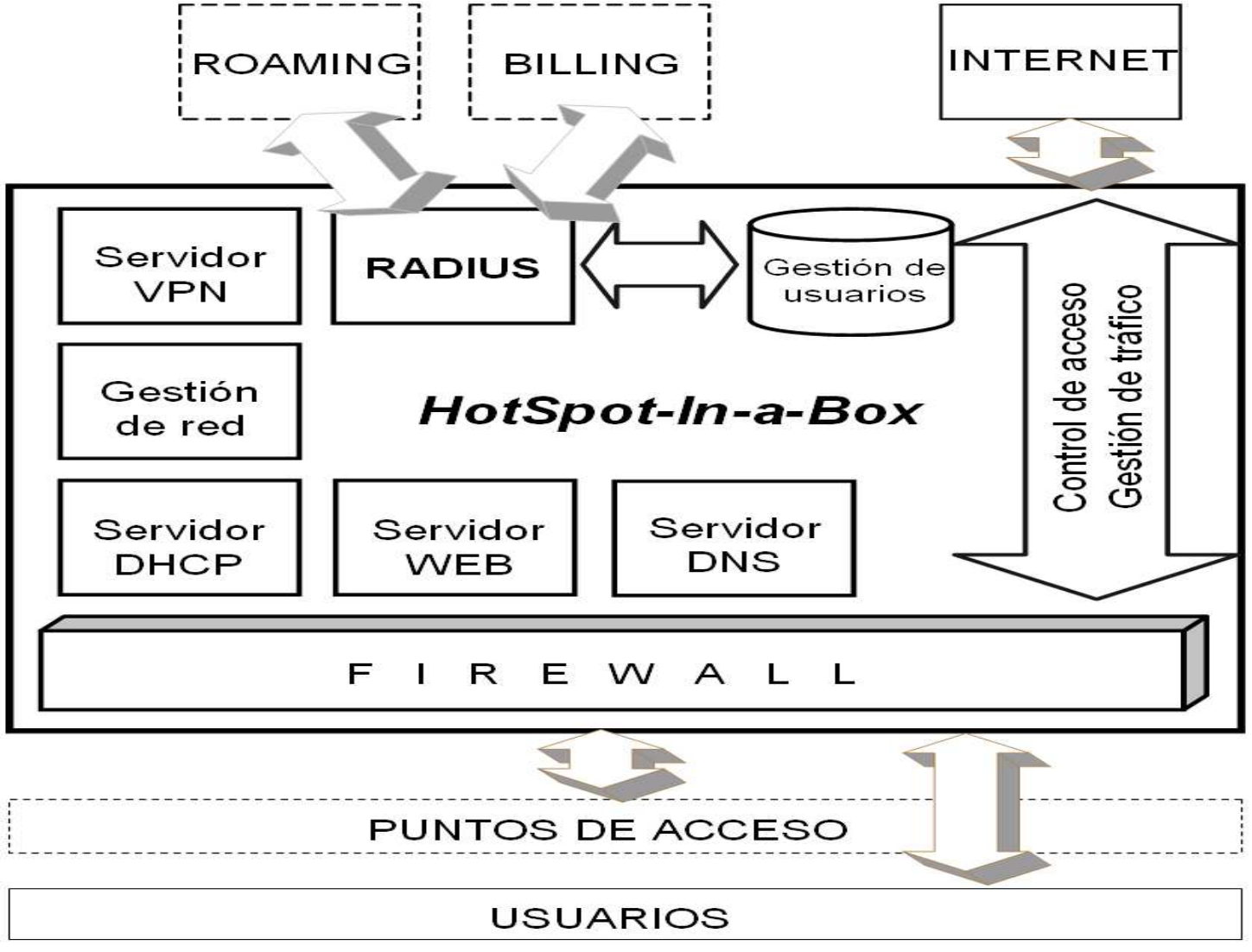
# Conceptos

- **Hotspot:** Lugar público o semi-público que dispone de conexión a Internet a través de tecnología *wireless*

  - Ejemplos: aeropuertos, cafeterías, hoteles, parques, salas de convenciones, recintos feriales, restaurantes, centros comerciales, bibliotecas...

- **WISP** (*Wireless Internet Service Provider*)**:** Operadora que ofrece servicios de conexión a Internet inalámbricos en estos lugares

# Diagrama de bloques



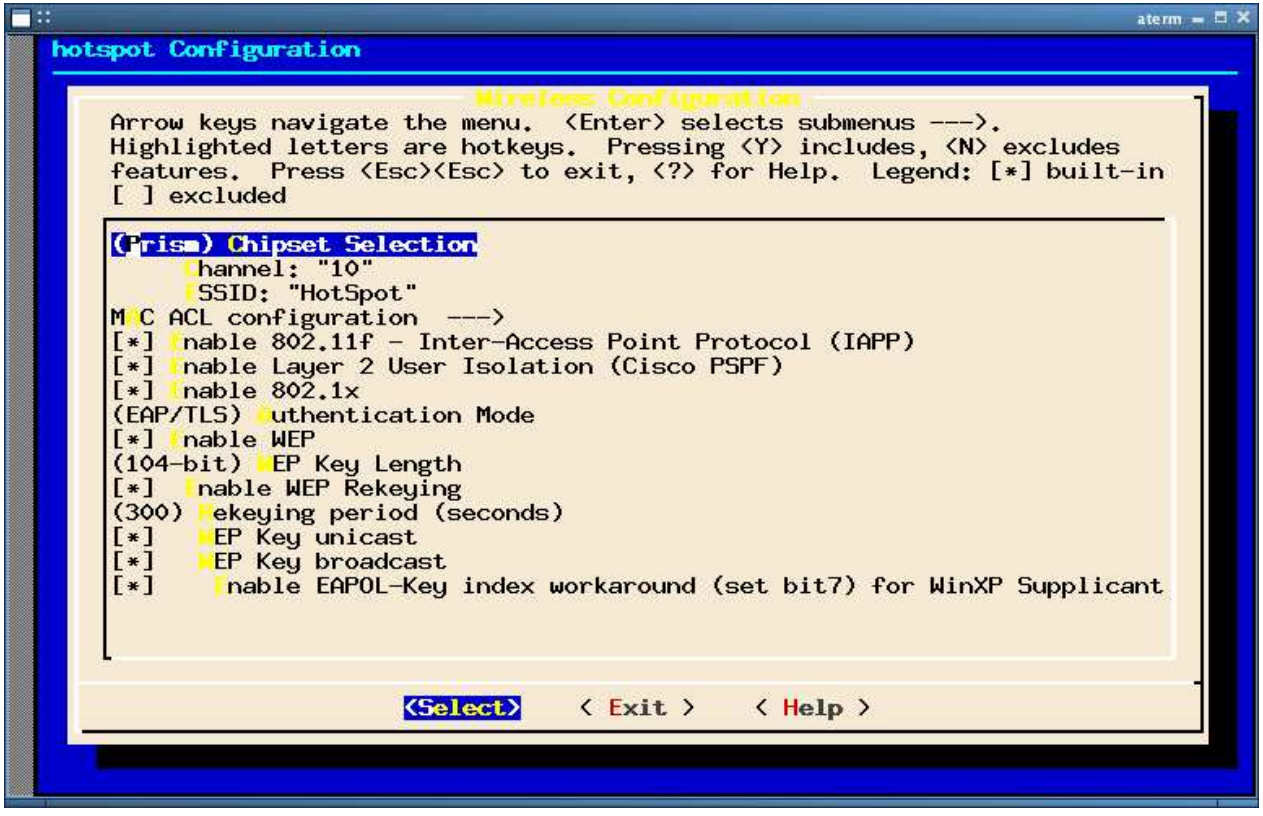
# Hardware Wi-Fi soportado

- Estándares 802.11 a/b/g
  - Auto-detecta *chipsets* Atheros y Prism
  - Velocidades de 11 y 54Mbps

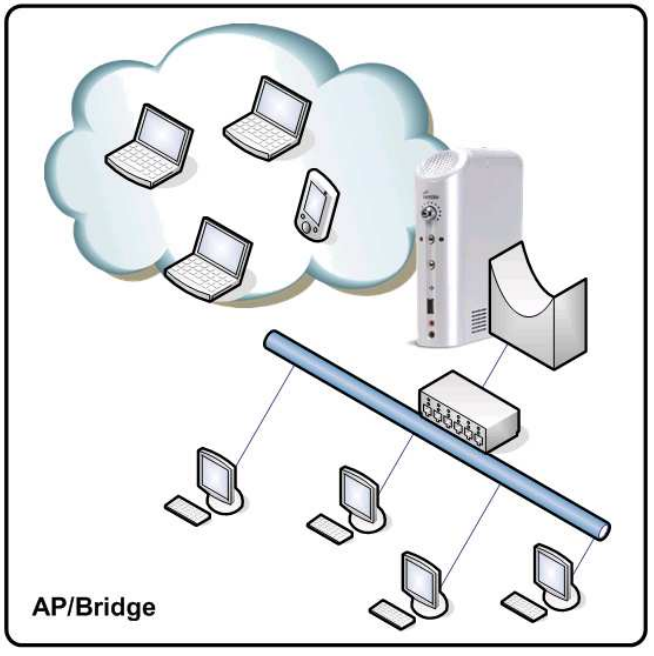

- AR5210 (802.11a)
- AR5211 (802.11b y 802.11g)
- AR5212 (802.11a, 802.11b y 802.11g)

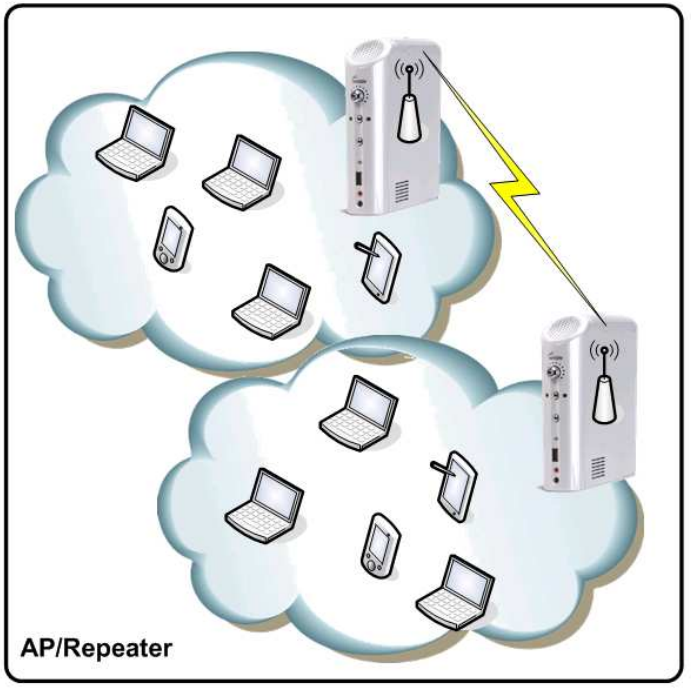

- Prism 2 (HFA3841 / HFA3842)
- Prism 2.5 (ISL3874)
- Prism 3

# Interfaz de configuración



# Modos de funcionamiento

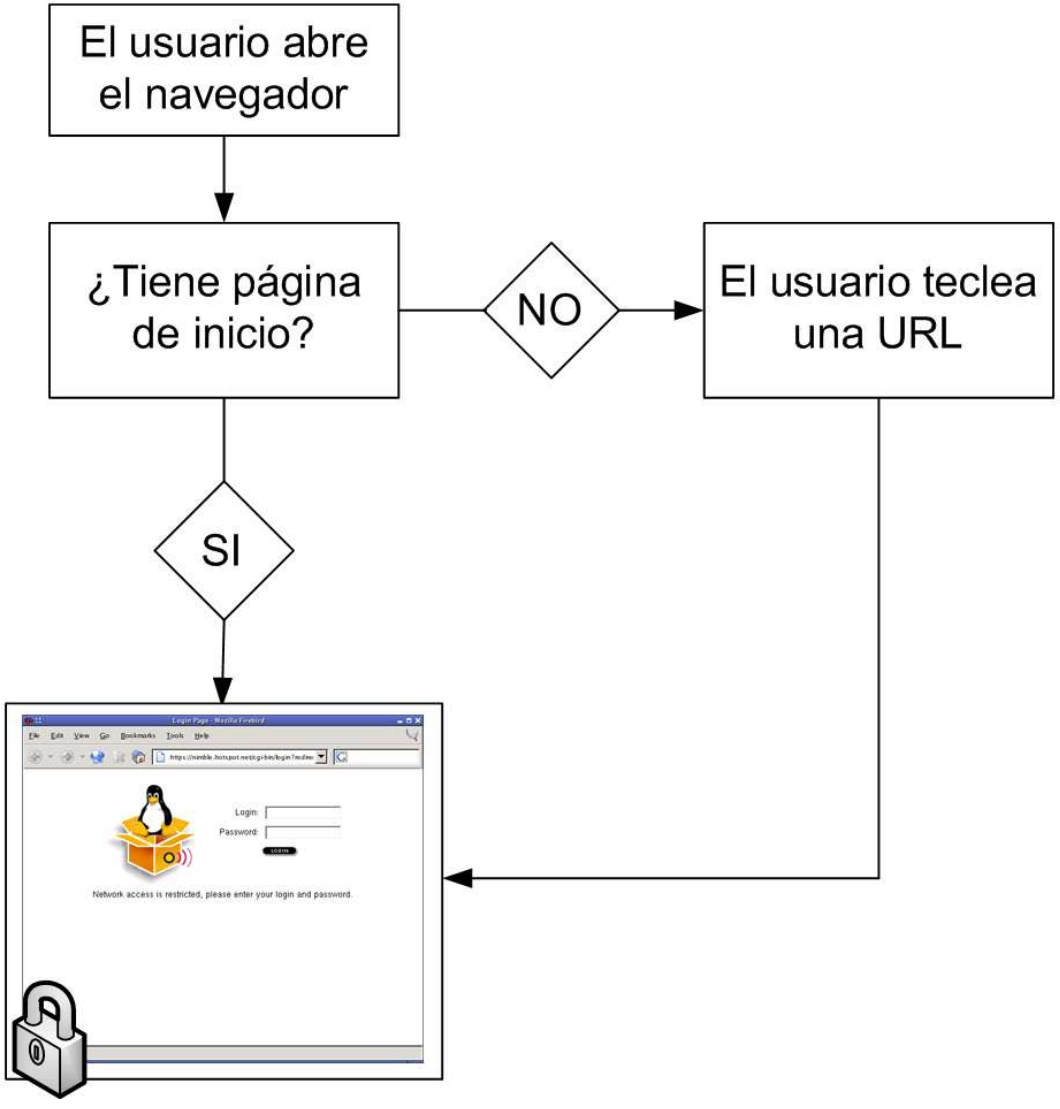



# Métodos de autenticación

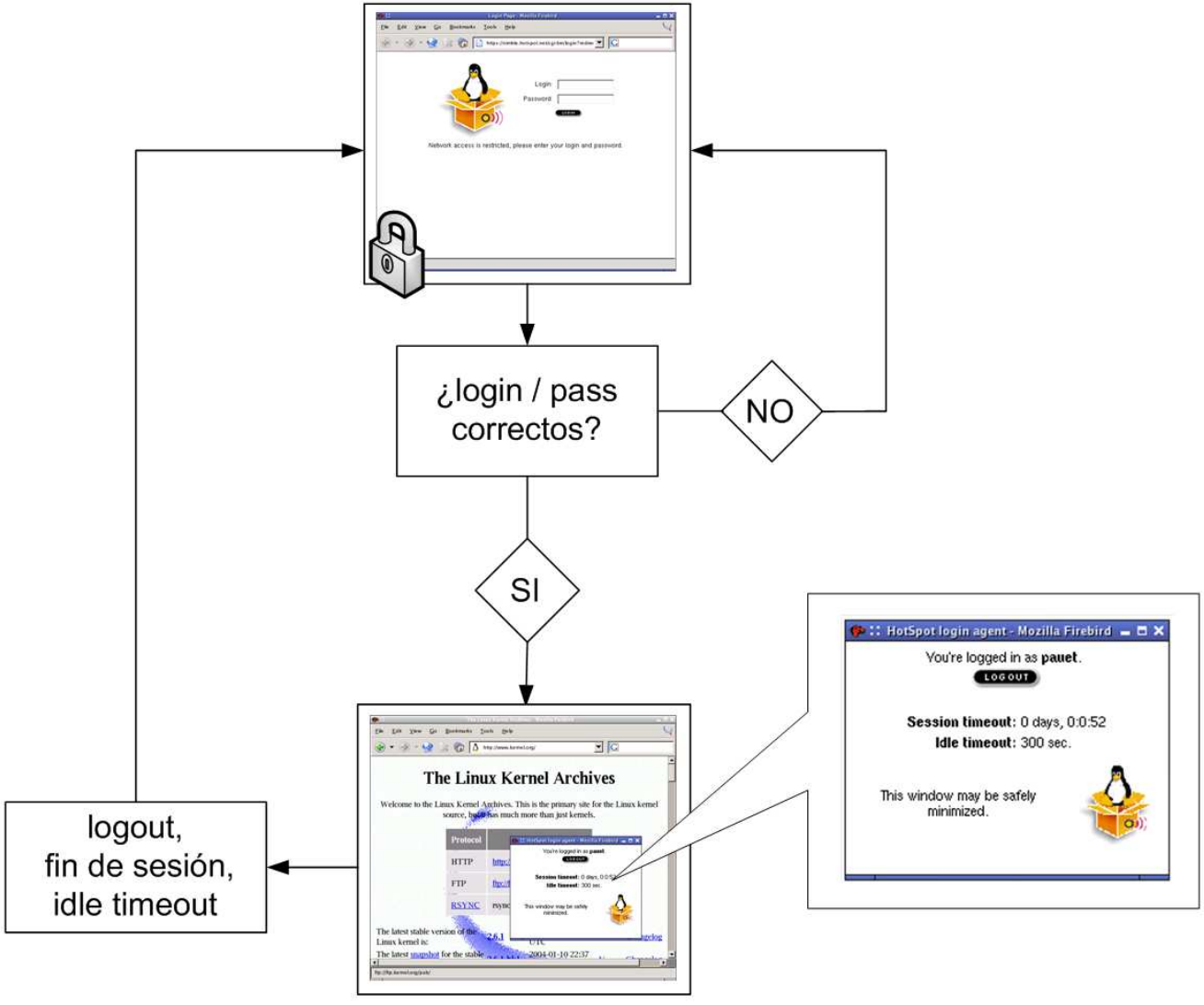
1. **Standard Web Access Method**
   - End User Experience

2. **IEEE 802.1x**
   - Componentes principales
   - Extensible Authentication Protocol over LAN
   - Mecanismos EAP
   - Proceso de Autenticación EAPoL
   - End User Experience

# Standard Web Access Method (I)



# Standard Web Access Method (II)



# 802.1x: Componentes principales

- **Supplicant**: La estación que requiere ser autenticada.
  - Responde a las peticiones hechas por el *Authenticator*.
- **Authenticator**: El dispositivo que hace posible que el *Supplicant* se autentique.
  - Controla el acceso físico a la red basado en el estado de autenticación del cliente.
  - Actúa como intermediario entre el cliente y el servidor de autenticación.
- **Authentication Server**: El dispositivo que proporciona el servicio de autenticación al *Authenticator*.
  - Determina si el *Supplicant* está autorizado para acceder a la red a partir de las credenciales que éste proporciona.

# 802.1x: Extensible Autentication Protocol over LAN

- El *Authenticator* y el *Supplicant* se comunican utilizando el protocolo EAP (RFC-2284).

# 802.1x: Extensible Autentication Protocol over LAN

- El *Authenticator* y el *Supplicant* se comunican utilizando el protocolo EAP (RFC-2284).
- 802.1x define un método para encapsular paquetes EAP sobre tramas Ethernet (EAP over LAN).

# 802.1x: Extensible Autentication Protocol over LAN

- El *Authenticator* y el *Supplicant* se comunican utilizando el protocolo EAP (RFC-2284).
- 802.1x define un método para encapsular paquetes EAP sobre tramas Ethernet (EAP over LAN).
- EAP soporta múltiples mecanismos de autenticación:

# 802.1x: Extensible Autentication Protocol over LAN

- El *Authenticator* y el *Supplicant* se comunican utilizando el protocolo EAP (RFC-2284).
- 802.1x define un método para encapsular paquetes EAP sobre tramas Ethernet (EAP over LAN).
- EAP soporta múltiples mecanismos de autenticación:
  - EAP/MD5 (Message Digest 5)
  - EAP/TLS (Transport Layer Security)
  - EAP/TTLS (Tunneled TLS)

# 802.1x: Mecanismos EAP

**EAP/MD5 (Message Digest 5)**

- Transmite nombre de usuario y hash de la contraseña
- NO realiza autenticación mutua

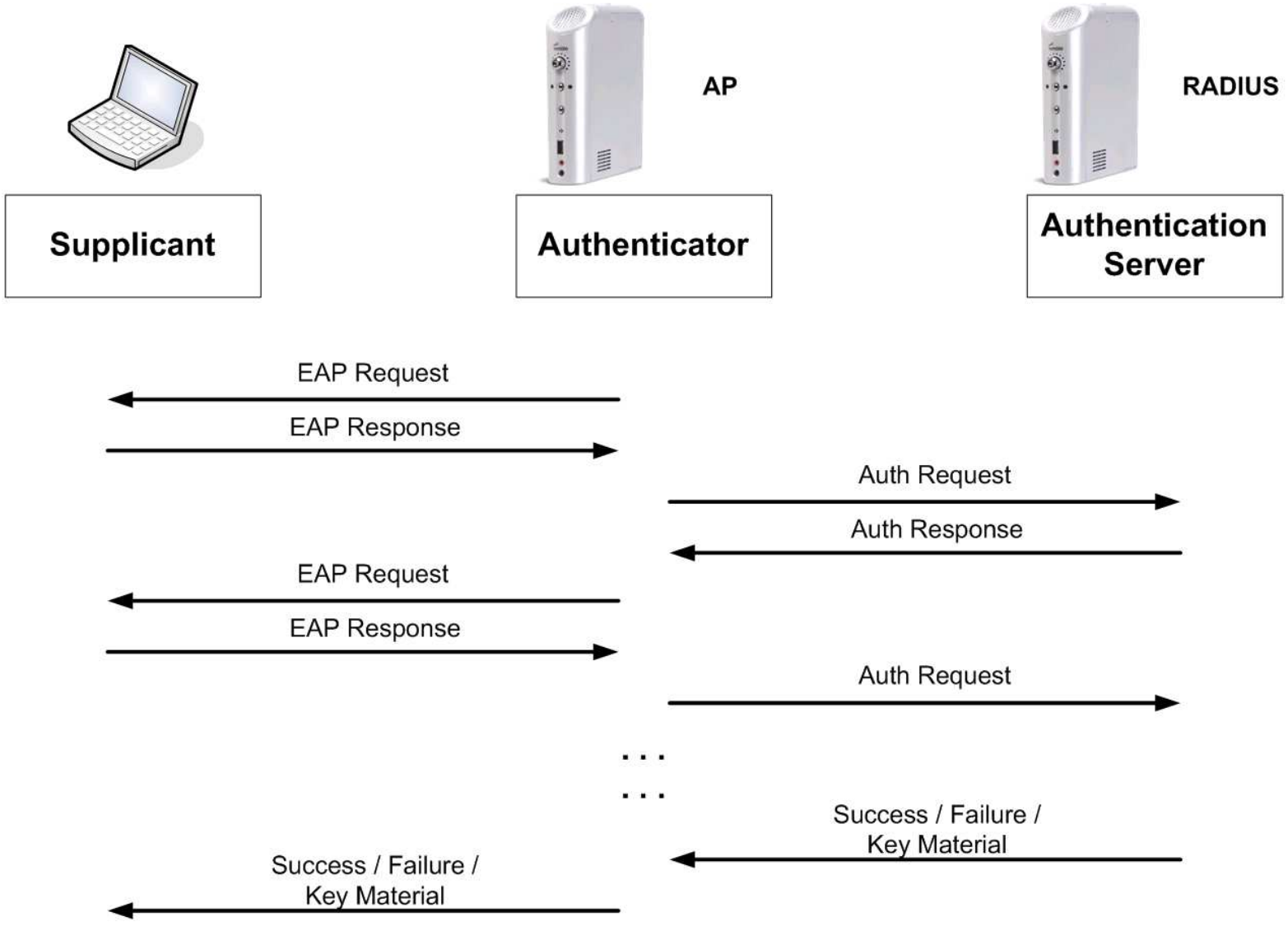
**EAP/TLS (Transport Layer Security)**

- Proporciona autenticación mutua → requiere certificados X.509 en ambas partes
- Establece un enlace seguro (TLS) después de la autenticación

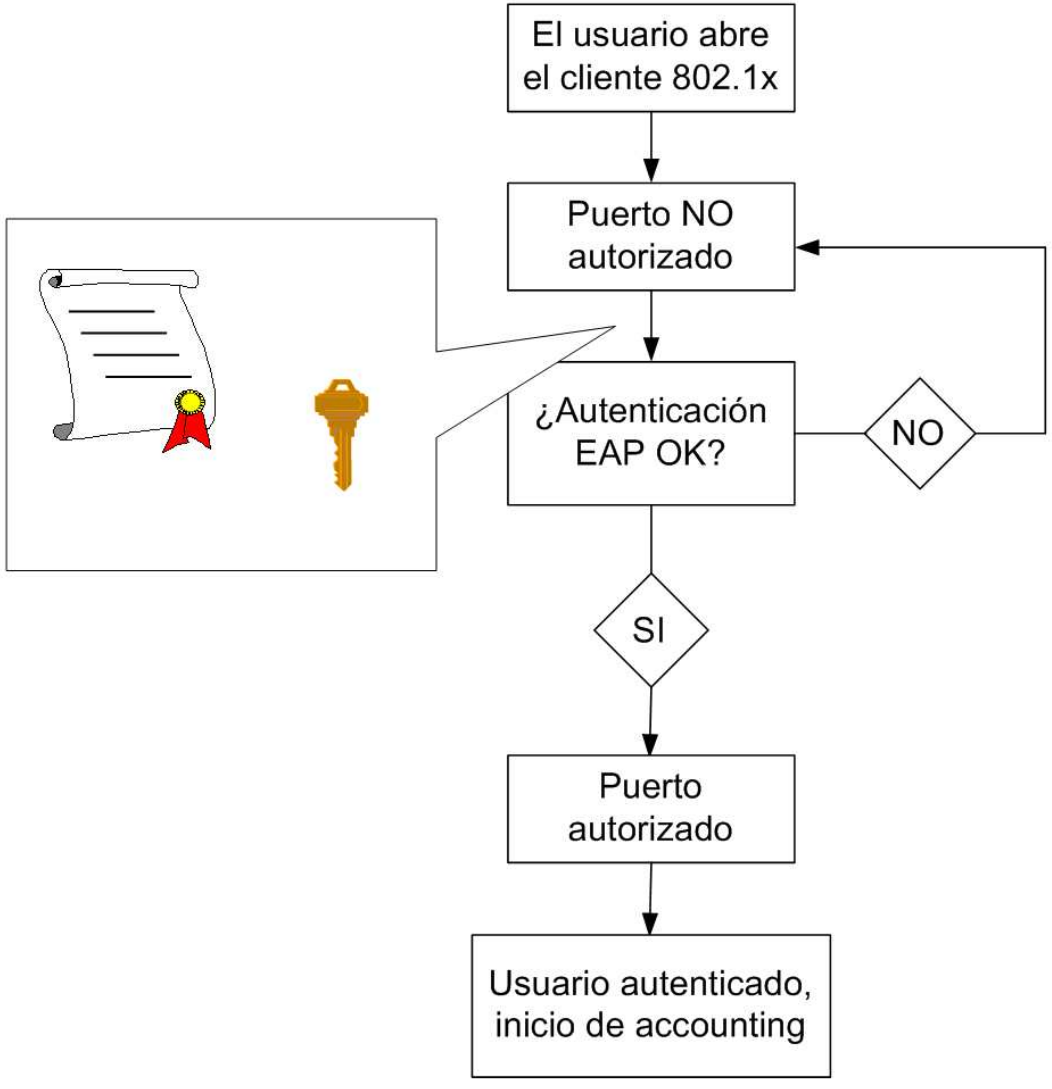
**EAP/TTLS (Tunneled TLS)**

- Autenticación mutua en dos fases
  1. El servidor de autenticación es autenticado al Supplicant utilizando un certificado X.509
  2. Se establece un canal seguro (TLS) a través del cual el Supplicant se autentica al servidor de autenticación

# 802.1x: Proceso de autenticación EAPoL



# 802.1x: End user experience

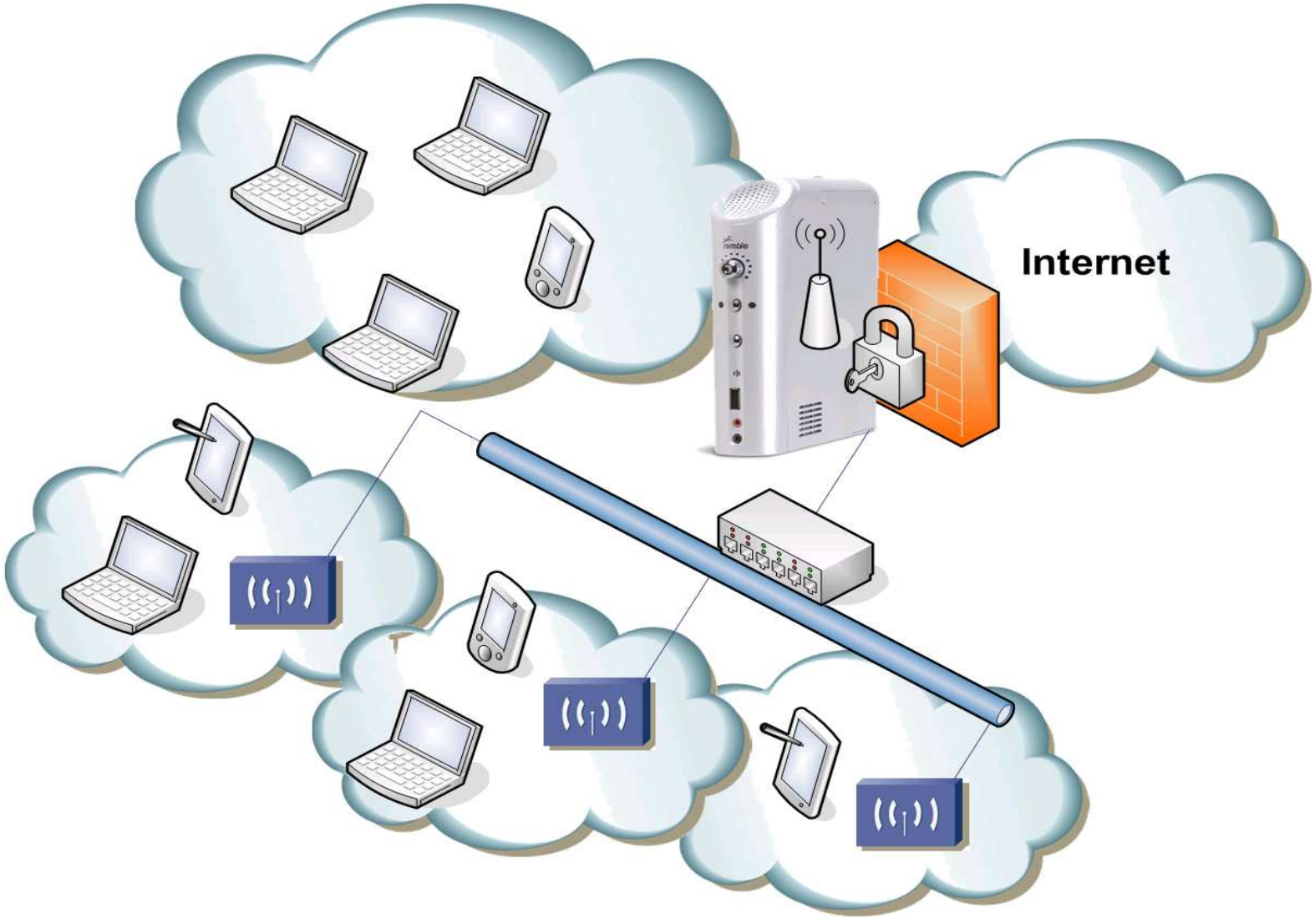


# Otras funciones del HotSpot-In-a-Box

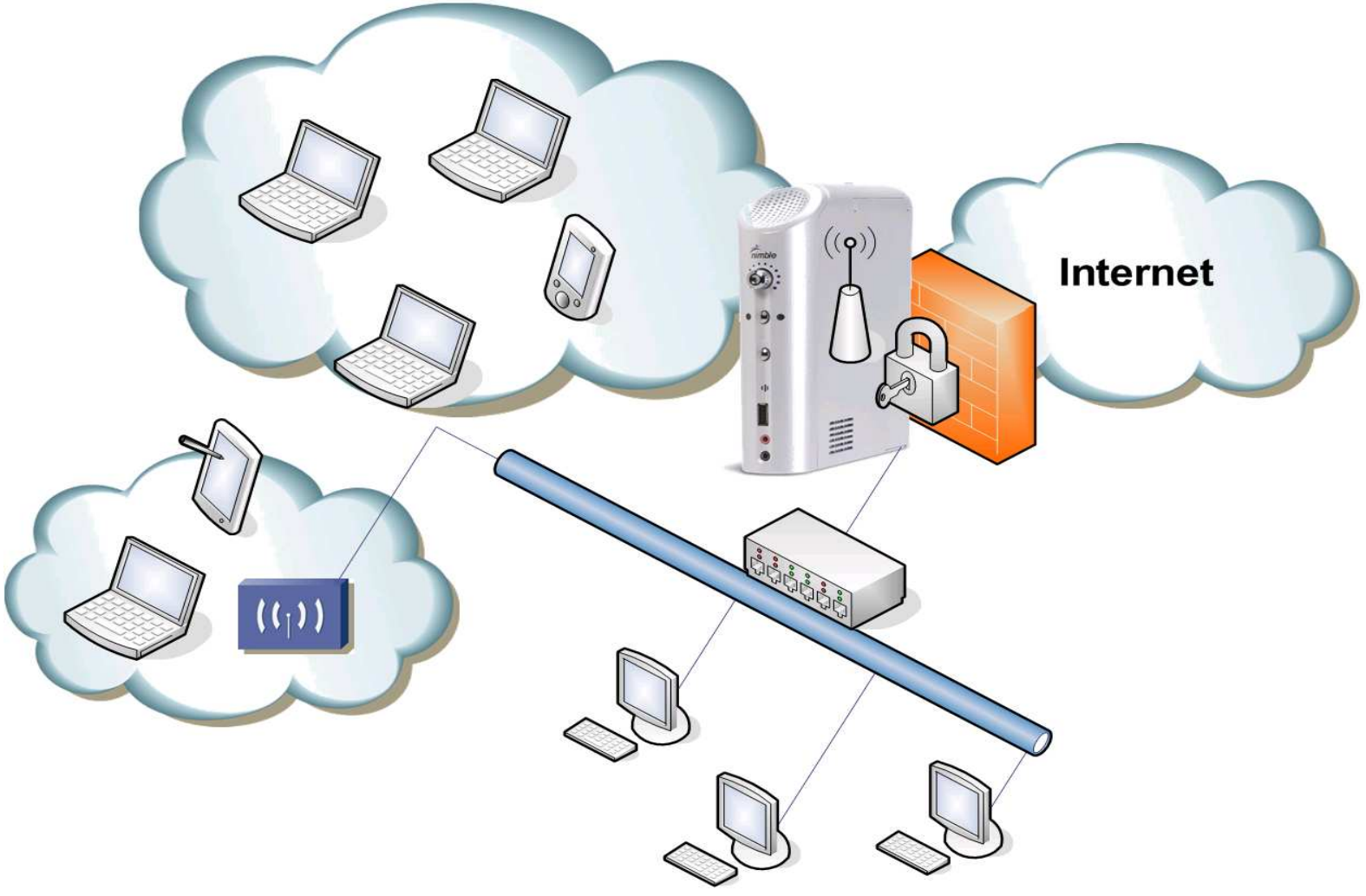
1. Gateway / Standalone HotSpot-In-a-Box
2. Access Point Controller
3. HotSpot with Desktop PCs
4. VPN passthrough
5. Multi-provider Roaming
6. Layer 2 user isolation

# Gateway / Standalone HotSpot-In-a-Box

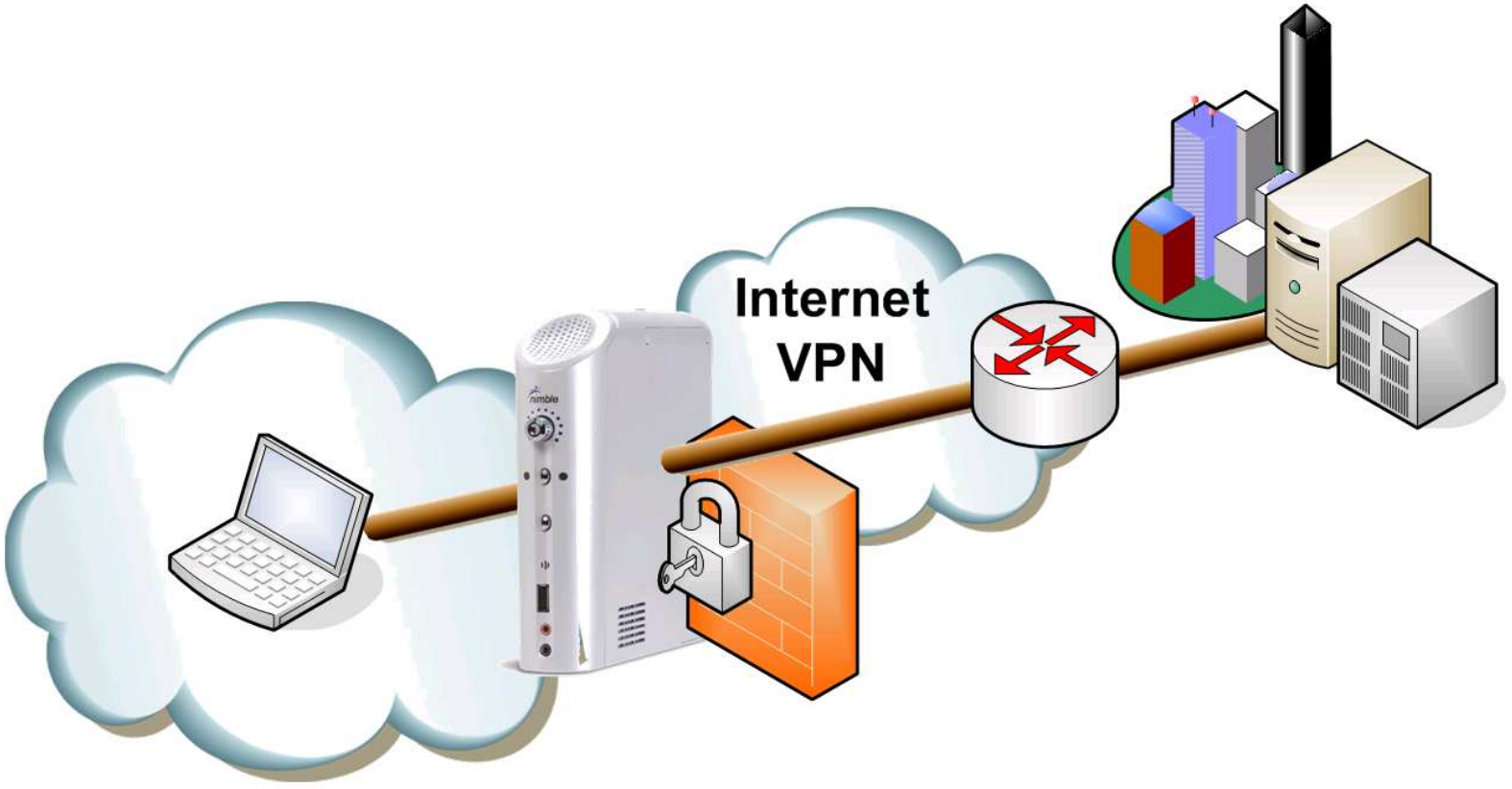
# Access Point Controller



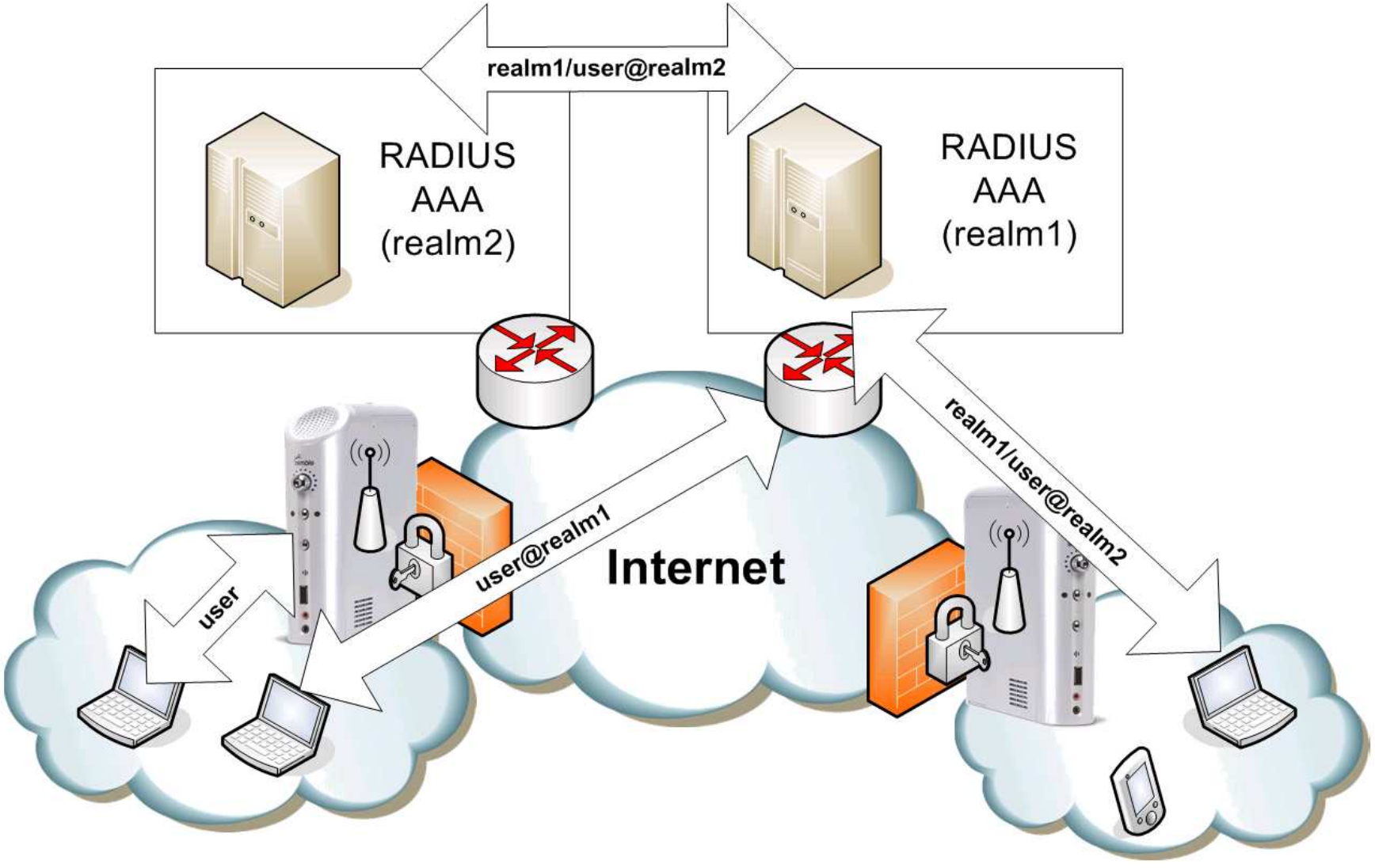
# HotSpot with Desktop PCs



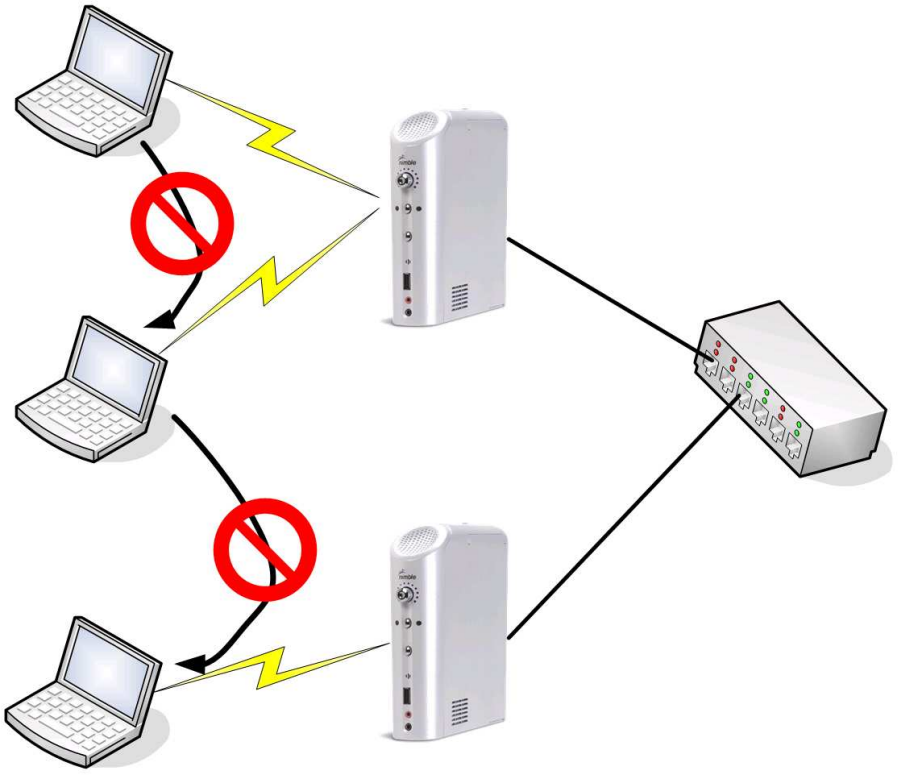
# VPN passthrough



# Multi-provider Roaming



# Layer 2 user isolation

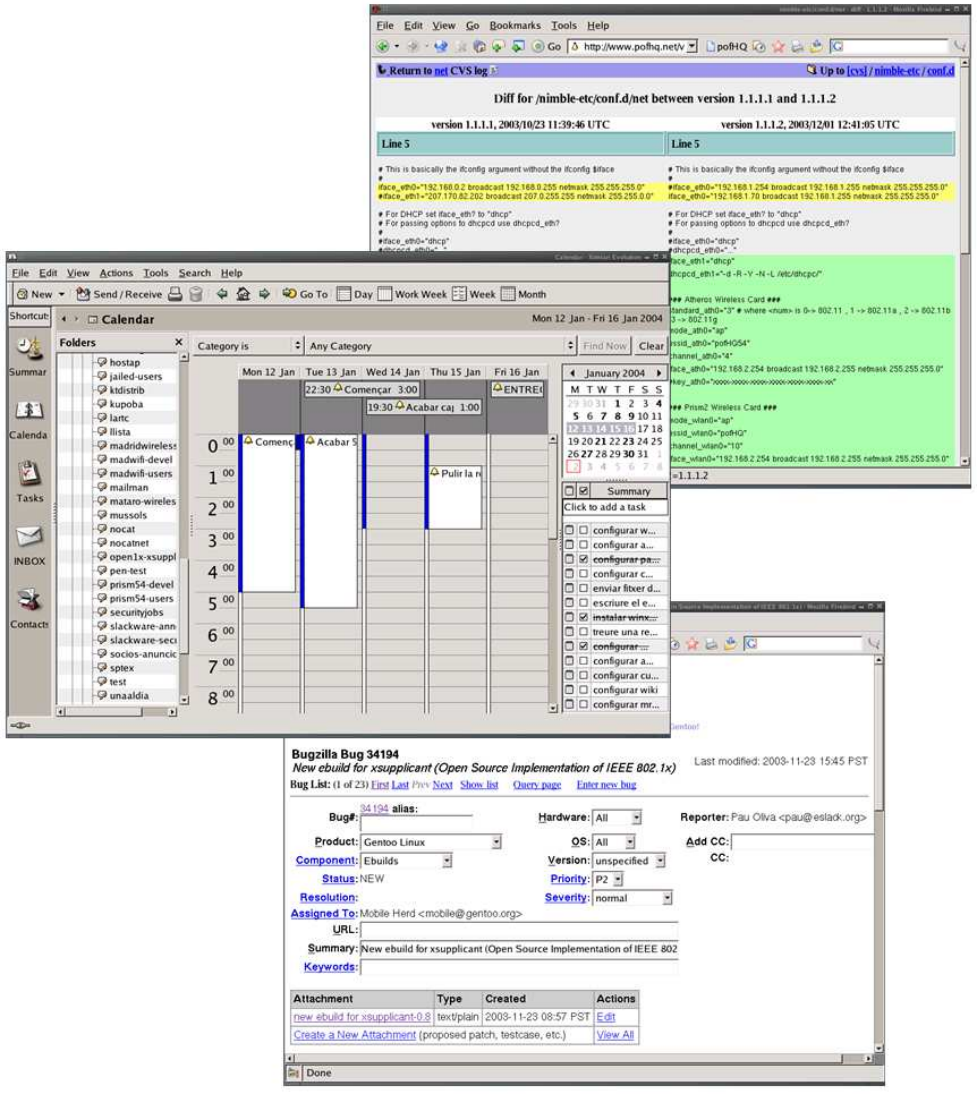


- Bloqueo de comunicación entre clientes:
  - Impide el acceso a unidades compartidas
  - Impide ataques

# Marco de Trabajo

1. **Entorno de desarrollo**
2. **Conclusiones**

# Entorno de desarrollo




| | |
|---|---|
| CVS | Vim |
| Evolution | LaTeX |
| Bugzilla | Gentoo |
| Bash | Perl |

# Conclusiones

- Se han conseguido implementar los mecanismos de control y seguridad previstos
- Se ha conseguido un resultado de características técnicas comparables a cualquier solución comercial equivalente
- Al tratarse de software libre todo el mundo puede contribuir en su desarrollo
- Es fácilmente adaptable a cualquier entorno de implantación
- Está basado en estándares abiertos
- Cumple con las necesidades tecnológicas del sector en la actualidad

# Demostración

## INSTRUCCIONES DE USO

- Abra su navegador de Internet. Aparecerá la página de bienvenida (si no aparece, por favor teclee la dirección de una página web cualquiera en la barra de direcciones).
- Introduzca su nombre de usuario y contraseña y pulse el botón “login”.

| login | password |
|---|---|
| unlimited | unlimited |
| user2m | user2m |
| user5m | user5m |
| user10m | user10m |

**White Page**: `http://www.kernel.org`