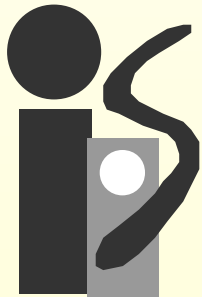
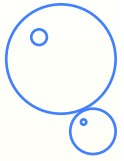
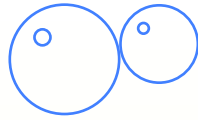


Conocimientos Estratégicos y de Control



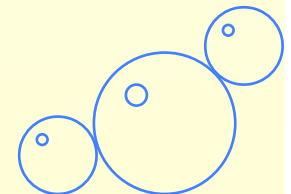
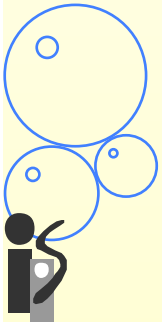
J.A. Bañares Bañares

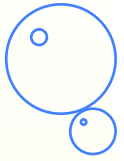
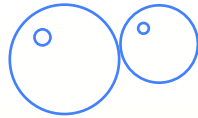
Departamento de Informática e Ingeniería de Sistemas
C.P.S. Universidad de Zaragoza



Objetivo

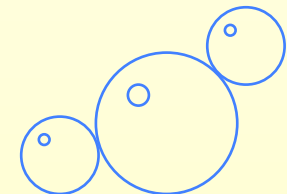
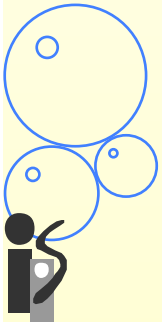
- **Objetivo:**
 - Modelos para la representación de razonamiento complejo.
 - La representación de los conocimientos de control
 - Soluciones que se plasman en arquitecturas software

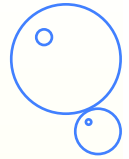
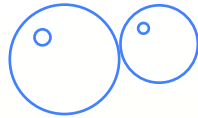




1. Conocimientos estratégicos

- **La función básica del control en los SBC:**
 - Manejar los conocimientos del dominio de manera que facilite y gestione su aplicación.
- **NO es suficiente con conocimiento y capacidad de inferencia.**
- **Hay que manejar los conocimientos y las inferencias como lo haría un experto en el dominio.**
 - Se precisa capacidad de aplicar las mismas **estrategias**, resolver las **excepciones** con los mismos criterios, reflejar las **incertidumbres** que afectan al razonamiento.

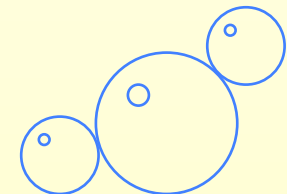
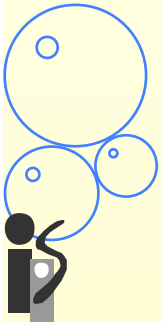




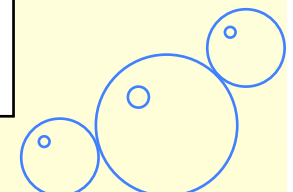
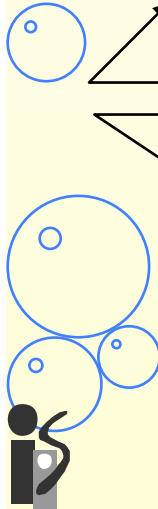
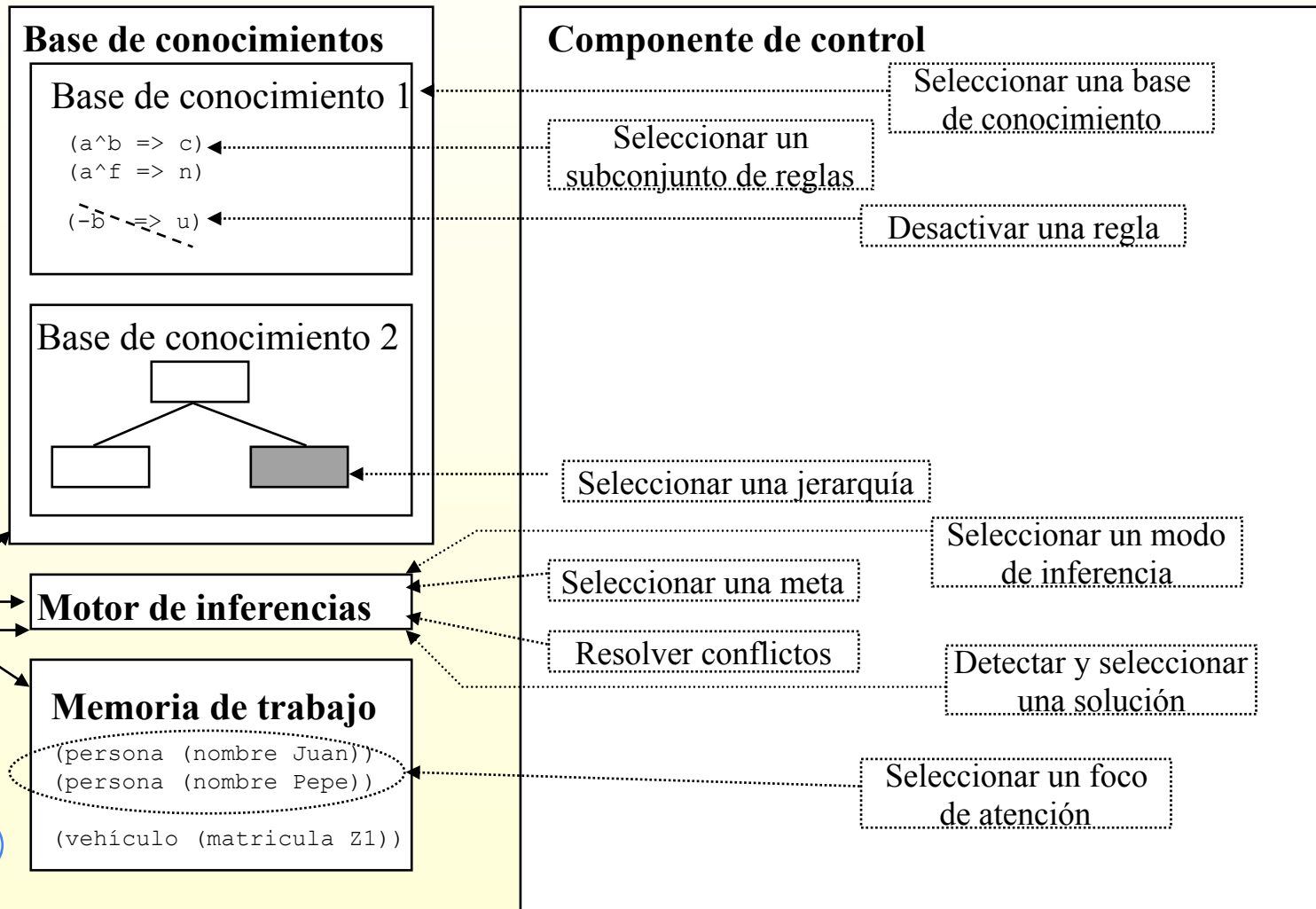
2. Componente de control

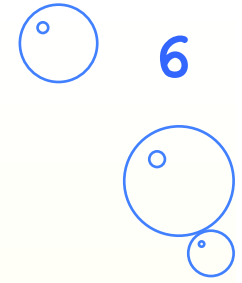
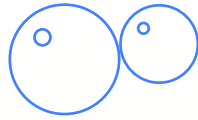
- **Componente de control de un SBC**

- Los criterios necesarios para decidir cómo aplicar los conocimientos del dominio durante la resolución del problema.
 - **Qué conocimientos** se deben utilizar en cada momento
 - **Cómo razonar** sobre ellos
 - **Dónde empezar y terminar** el proceso de razonamiento
 - **Resolver conflictos** que puedan aparecer
 - Evaluar soluciones alternativas
 - Determinar cuando están resueltos determinados subproblemas
 - Decidir si se interrumpe la resolución de un subproblema y se comienza la resolución de otro
 - Qué datos usar como entrada



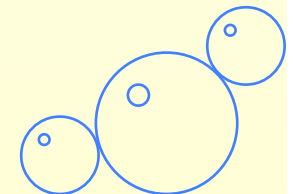
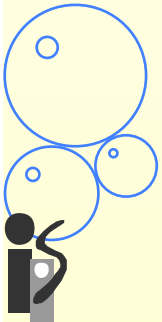
Tareas típicas del control



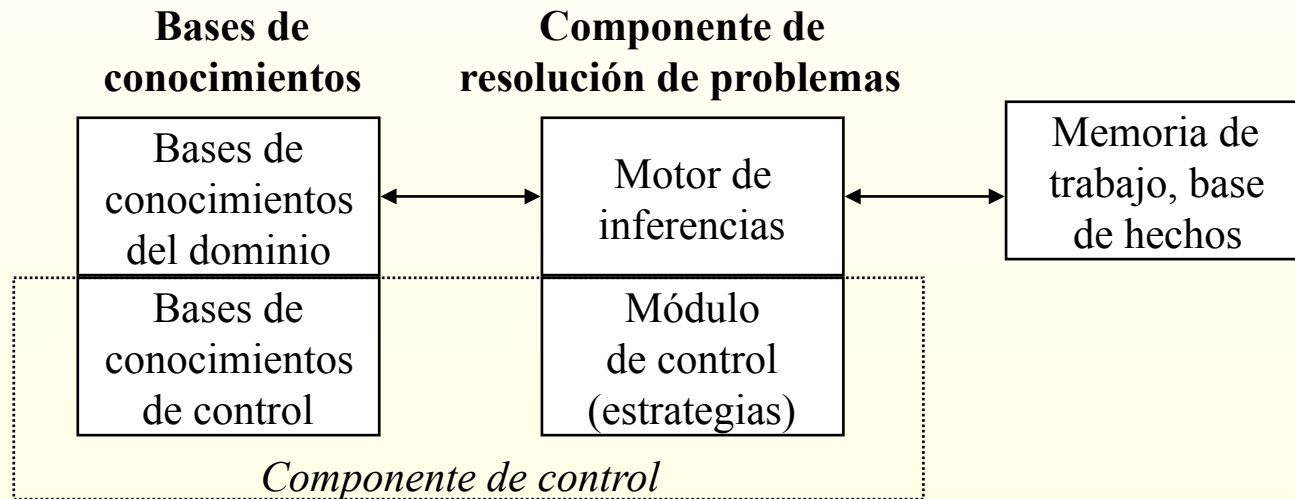
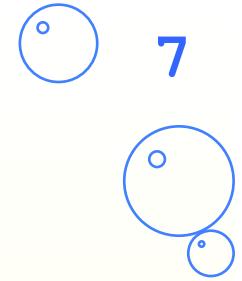


Componente de control

- **Componente control = motor inferencia + módulo control**
- **El motor de inferencia:**
 - Es un programa de propósito general
 - (demostrador lógico de teoremas, un equiparador de patrones, procedimientos de búsqueda en redes, etc.)
 - No es capaz de resolver los conflictos que surjan, ni de identificar soluciones o establecer metas.
- **El módulo de control:**
 - Incorpora una estrategia de resolución de problemas especialmente seleccionada para el tipo de problemas



Localización del componente de control

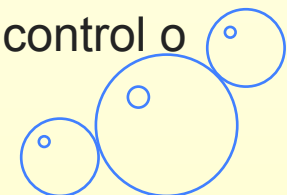
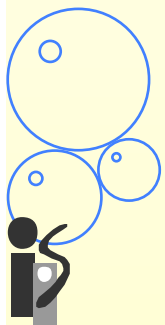


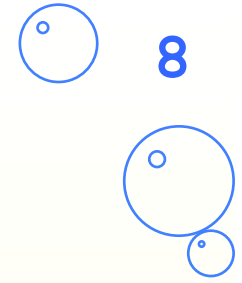
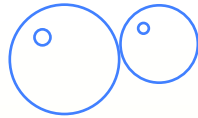
- **Implícito:**

Integrado como parte del programa que implementa la estrategia de control o estrategia de resolución del problema dentro del módulo de control

- **Explícito:**

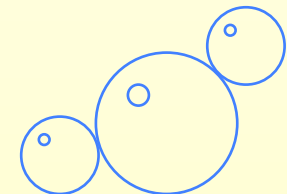
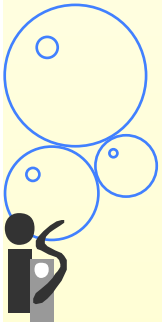
En algún lugar de las bases de conocimiento. Conocimiento de control o metaconocimiento.

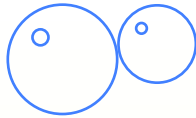




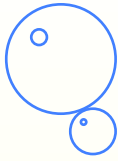
Tipos de control

- **Según la posibilidad de revisar y modificar las decisiones de control:**
 - Control irrevocable: Una vez tomada una decisión es irrevocable.
 - Control típico de los programas algorítmicos
 - Es necesario prever a priori todas las posibles situaciones
 - Control provisional:
 - Existe la posibilidad de volver atrás si en un momento dado nos damos cuenta de las decisiones adoptadas no fueron adecuadas.
 - Backtracking cronológico
 - Backtracking basado en justificaciones



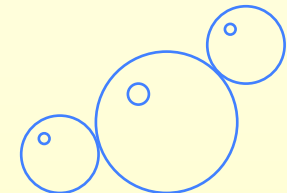
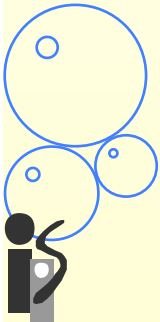


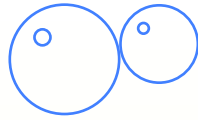
Tipos de control



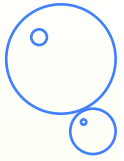
- **Según la forma de representar el conocimiento de control:**

- Control *paramétrico*: Característico del control de bajo nivel
 - Parámetros que especifican informaciones relevantes para el control como la prioridad de reglas
- Control *declarativo*: Representados de forma declarativa en alguna base de conocimiento de control, o dentro de bases de conocimiento del dominio.
 - Sus decisiones suelen estar referidas a la elección de la estrategia de resolución del problema. (metarreglas)
- Modelos de control: **Modelos que facilitan resolver problemas en los que no exista una estrategia de solución a priori.**
 - Ejemplo: *Arquitecturas* de pizarra



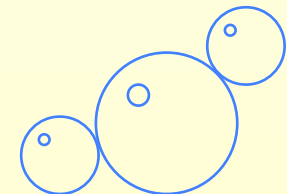
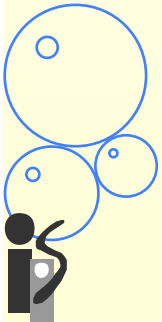


Tipos de control

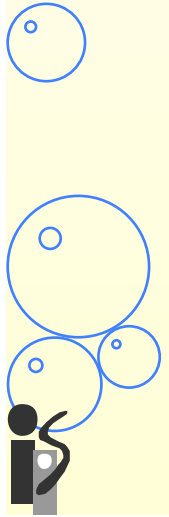
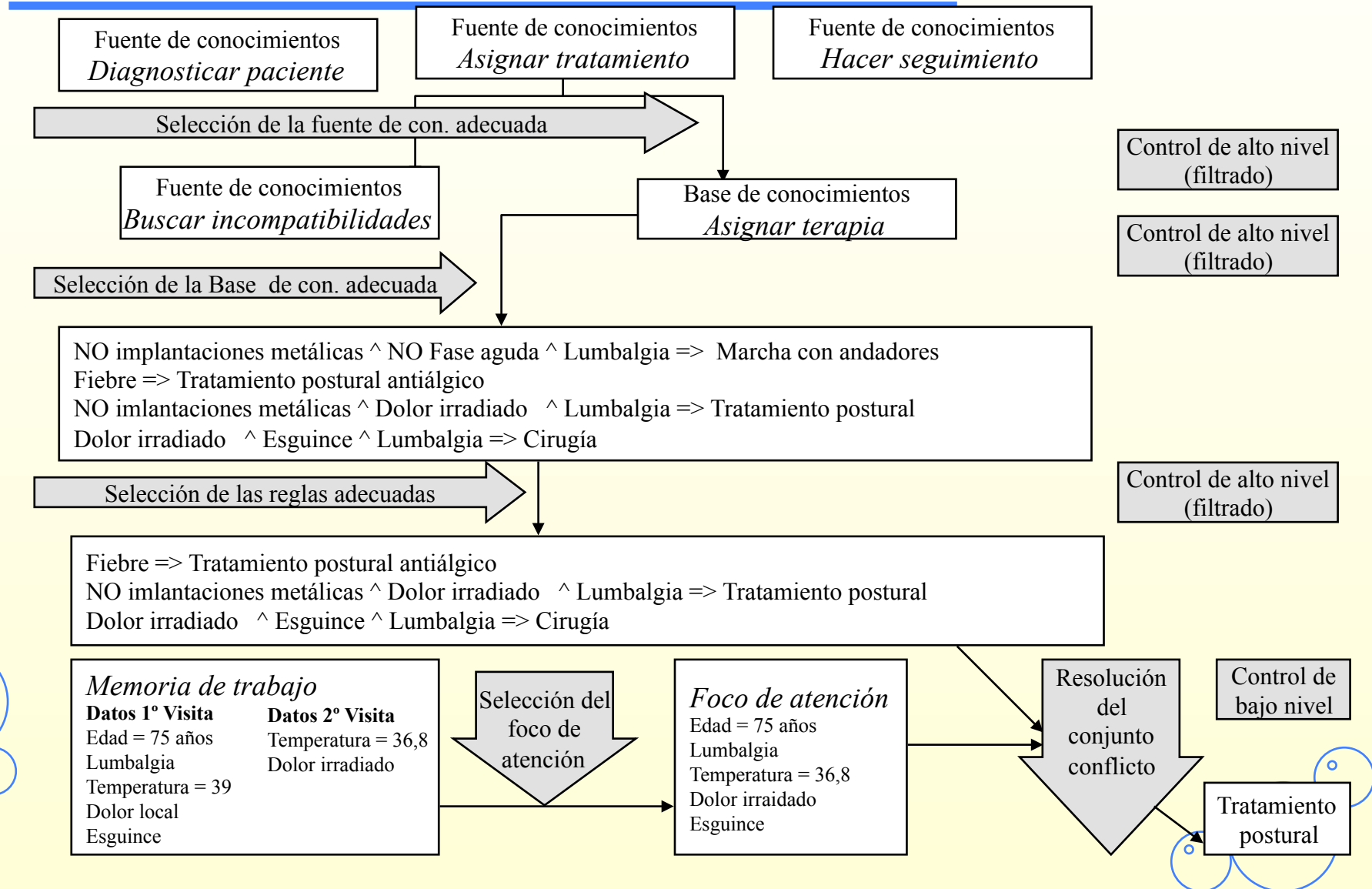


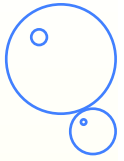
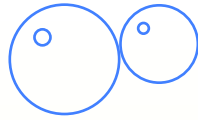
- **Tipos de decisiones de las que se encarga el control**

- Tomando la fase de selección dentro del ciclo de reconocimiento/actuación de un sistema de producción se distinguen las siguientes etapas:
 - 1. Fase de filtrado: Selección de un subconjunto de reglas.
 - 2. Fase de Reconocimiento: Obtención del conjunto conflicto. Precisamente se establece también el foco de atención (datos de la memoria de trabajo con los que se hará el reconocimiento).
 - 3. Fase de resolución del conjunto conflicto: Se selecciona una regla.
- “Suele decirse”
 - Control de *alto nivel* cuando está asociado a la etapa de filtrado
 - Control de *bajo nivel* en la fase de resolución
 - Por lo general, cuando es una decisión estratégica (cambio de objetivo o subproblema) se considera alto nivel, y el resto bajo nivel.



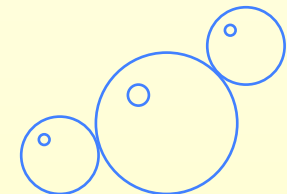
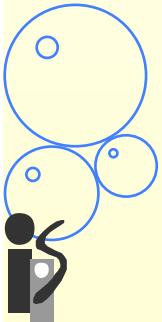
Ejemplo de intervención de diferentes tipos de control

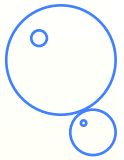
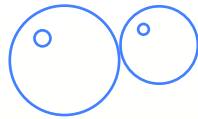




Control de bajo nivel

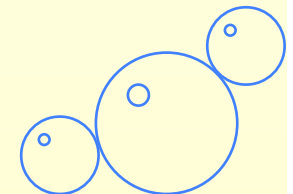
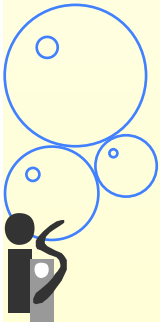
- **Sistemas de producción:**
 - Especificidad, novedad, refracción, prioridades (control global)
- **La búsqueda como control:**
 - Las estrategias de búsqueda no son más que un tipo de control de bajo nivel
 - Afectan fundamentalmente a la **fase de creación del conjunto conflicto** y a la **selección del foco de atención**.
 - Generar y verificar, Escalada, búsqueda en profundidad y anchura, búsqueda primero el mejor.

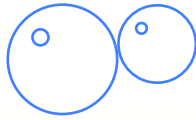




Control de alto nivel

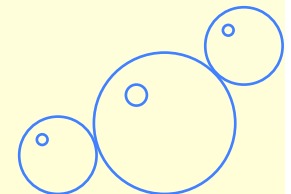
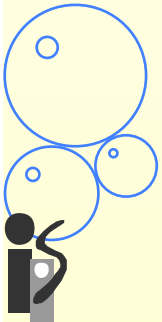
- **La complejidad del problema puede suponer que los mecanismo de bajo nivel no sean suficientes**
 - Se requiere la adopción de estrategias específicas para la organización de este tipo de sistemas.
- **Problemas planteados por los problemas complejos**
 - Manejo de *grandes cantidades de conocimientos*
 - Opciones: - Representar todo en un único módulo y utilizar control de bajo nivel para filtrar.
 - Definir módulos separados
 - *Combinación de varios formalismos.*
 - Se puede utilizar módulos con diferentes formalismos y encontrar la forma de combinarlos.
 - Representación y manejo de *estados complejos*
 - Dificultad de representar un estado

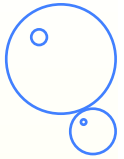
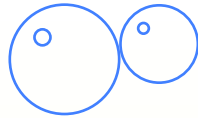




Simplificación del manejo del espacio de búsqueda

- Formas de abordar la representación de estados complejos
 - 1. Encontrar la manera de pasar de un estado a otro sin que sea necesario recalcular totalmente el nuevo estado.
 - 2.. Seleccionar en cada momento una parte del estado en la que centrar el trabajo (un foco de atención).
 - 3. Descomponer a priori el problema original en subconjunto de subproblemas más pequeños y sencillos de resolver.





3. Control de alto nivel

- **Siempre se utiliza alguna forma de representación de conocimientos de control explícita**

- Mediante alguno de los formalismos habituales (**metarreglas**)

- **Acciones que realizan las metarreglas**

- Acciones que afectan al sistema en su totalidad

- Parar/Activar, Esperar entrada

- Acciones que actúan sobre reglas individuales

- Inhibición/deshinibición, Modificación de su prioridad, coste, filtrar

- Acciones que actúan sobre el control de bajo nivel

- Modificación de los criterios de resolución del conjunto conflicto

- Modificación sobre el modo de razonamiento

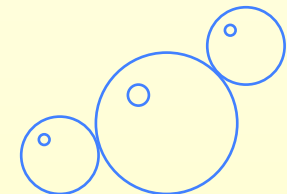
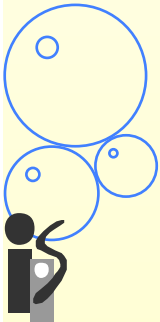
- Acciones que actúan sobre conjuntos de metas:

- Añadir / eliminar metas, Reordenar, filtrar

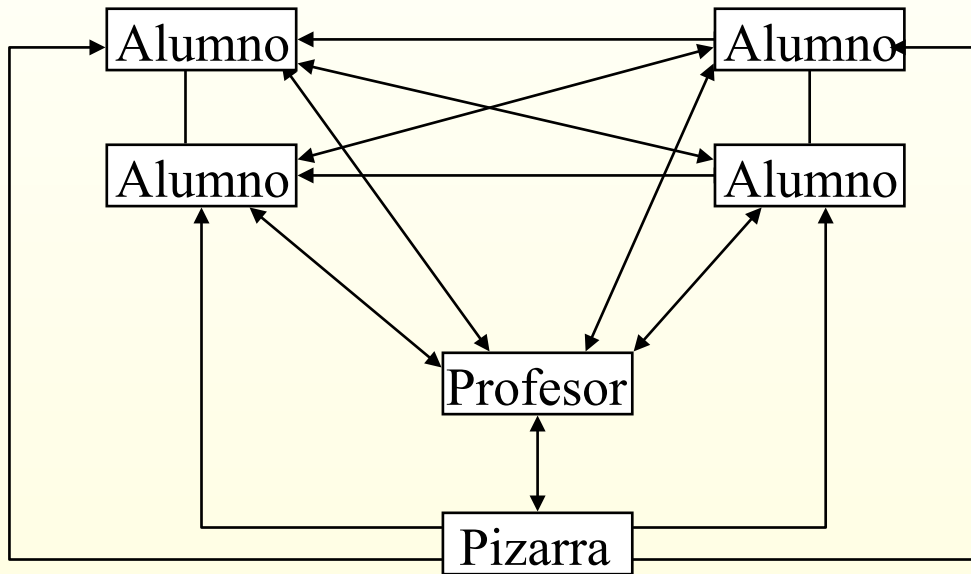
- Acciones que actúan sobre bases de conocimientos

- Añadir/eliminar reglas, reordenar, filtrar

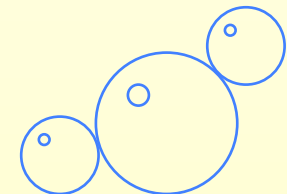
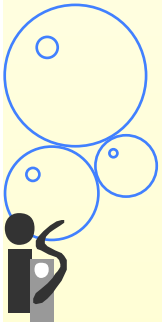
- Mediante la aplicación de modelos de control que dotan al sistema de una arquitectura característica (pizarras)

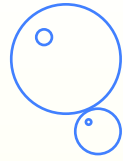
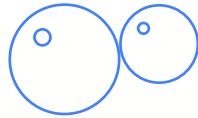


Pizarra en el aula

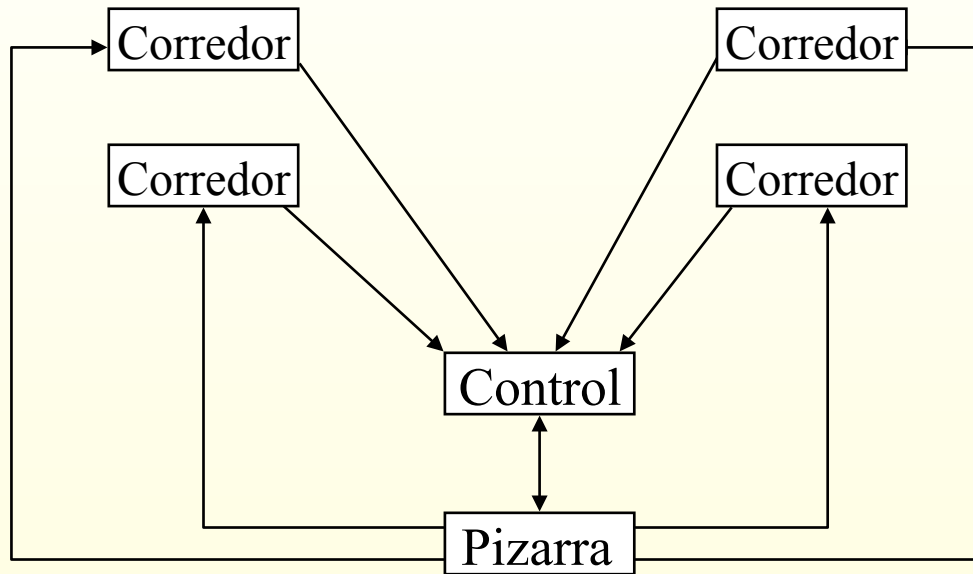


- Características para un buen funcionamiento:
 - Pocos alumnos, con capacidades similares
 - En una misma habitación, con buen conocimiento entre sí
- En el momento que se incumpla alguna de estas características toma relevancia la pizarra

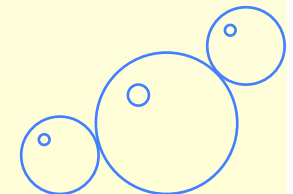
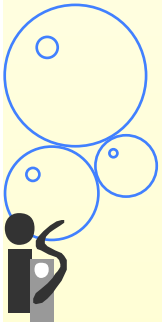


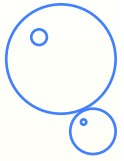
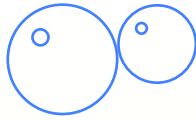


Pizarra en el mercado de valores

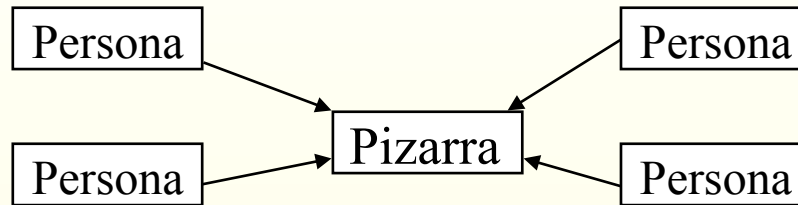


- Características de funcionamiento:
 - Es innecesaria la reunión física de los actores
 - Funciona aunque se modifique el comportamiento de los actores, se añadan o se eliminen actores
- El papel del control es simplemente tramitar las operaciones y anotarlas

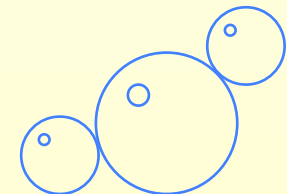
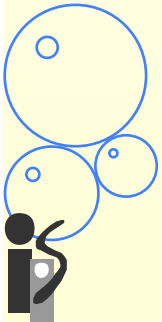




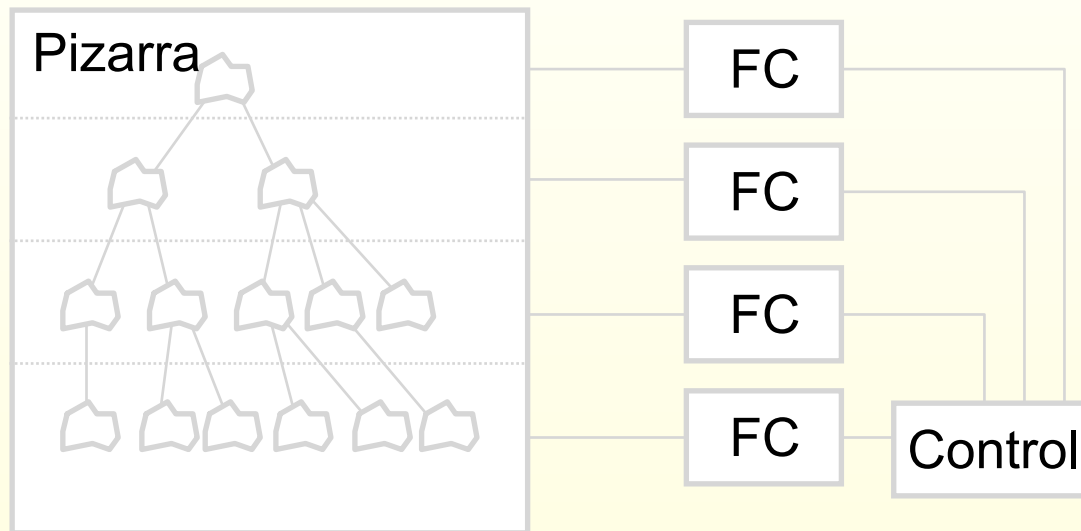
Resolución del problema del puzzle



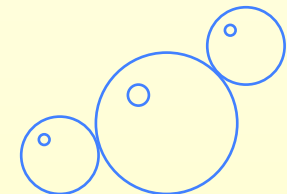
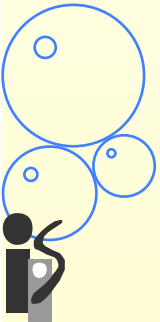
- Características de funcionamiento:
 - Cada persona tiene piezas de un puzzle
 - El rompecabezas puede resolverse en completo silencio

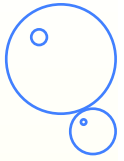
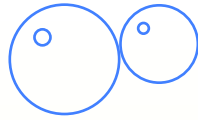


Arquitectura de pizarra



- Necesidad del control
 - Sólo puede actuar una FC cada vez
 - La pizarra es una estructura de datos compartida

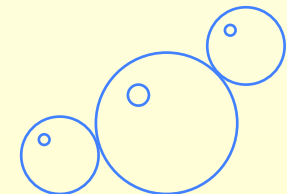
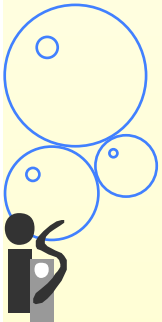


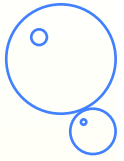
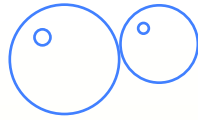


4. Arquitecturas Cooperativas

- **Arquitectura Cooperativa**

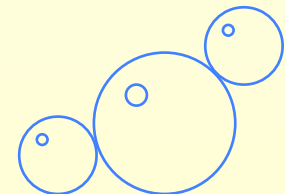
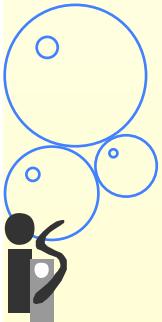
- Una clase particular de arquitecturas **multiagente**
- Ideadas para facilitar la colaboración entre agentes
 - Para resolver un problema común
 - Para mejorar la eficacia de cada uno en la tarea que tenga encomendada
- Los componentes básicos que forman la arquitectura de un sistema cooperativo, distribuido o no, son:
 - Un **coordinador** de actividades a realizar
 - Un **comunicador** para intercambiar información entre módulos
 - Un **sistema de mantenimiento de la verdad distribuido** para garantizar que todos los sistemas son consistentes en los puntos en los que se requiere cooperación.

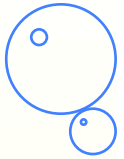
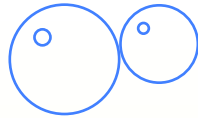




Tipos de arquitecturas

- **Arquitecturas con coordinación asistida**
 - Los agentes delegan funciones a programas especiales para conseguir la comunicación
- **Arquitecturas con comunicación directa**
 - Los agentes manejan su propia coordinación

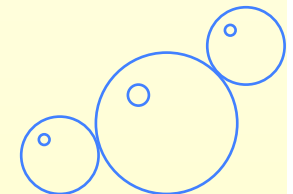
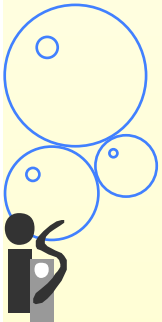




4.1 Con coordinación asistida

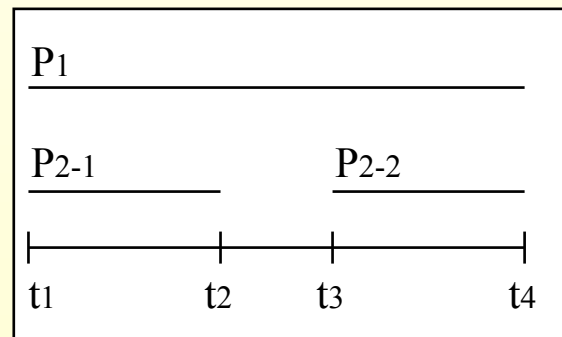
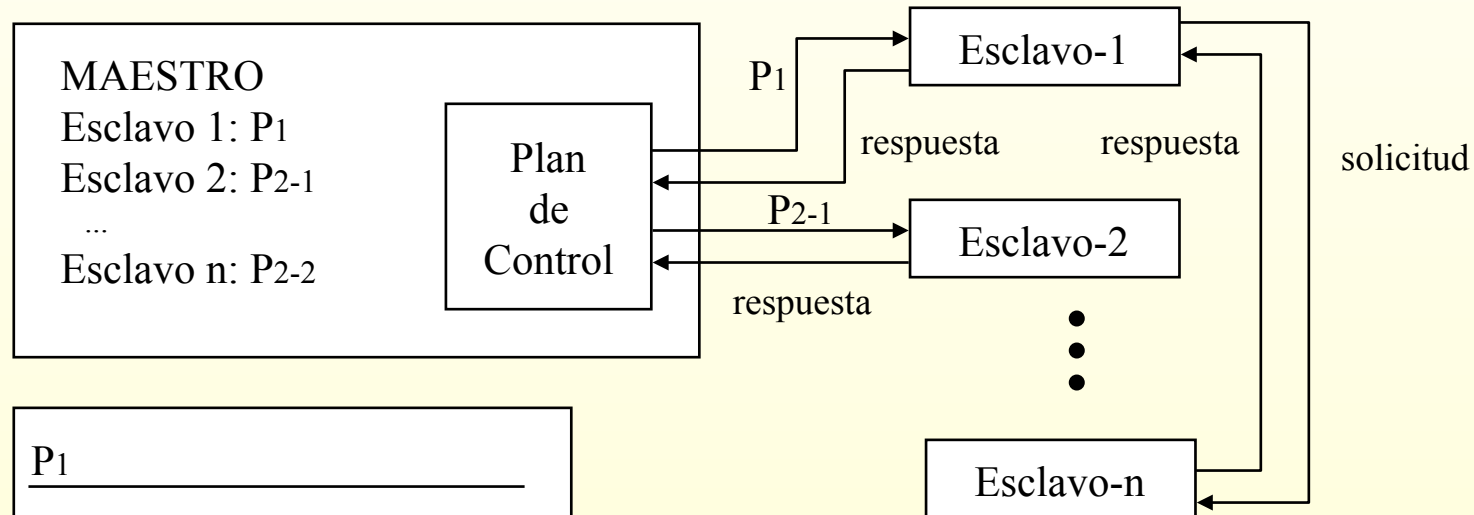
- **EL modulo principal es el coordinador**

- El IC define el comportamiento del coordinador en función de las características del problema
- Consta de:
 - *Un planificador* que descompone un problema en subproblemas mutuamente independientes (Planning)
 - *Un asignador* que distribuye las tareas a realizar entre los agentes subordinados (Scheduling).
- Garantiza que los agentes que forman el sistema cooperativo trabajan conjunta y eficientemente
 - a) Conoce la función que realiza cada uno de sus agentes subordinados y sus capacidades
 - b) Dispone de un plan de control con los planes a realizar, dependencias entre ellos, y su secuenciamiento
 - c) Se asegura de que los subplanes se finalizarán lo antes posible
 - d) Una vez que la tarea se ha distribuido es necesario sincronizar los agentes subordinados, a menos que las tareas sean independientes.

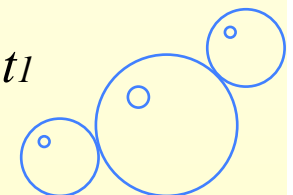
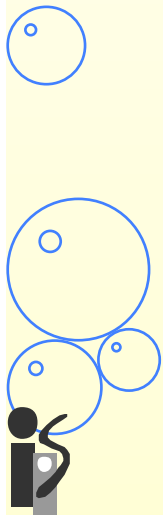


Coordinación asistida con maestro

- El agente maestro está encargado de hacer el plan y distribuir los subplanes a sus agentes "esclavos", que llevan a cabo la tarea y devuelven los resultados al maestro.**
 - Los agentes esclavos pueden comunicarse entre sí



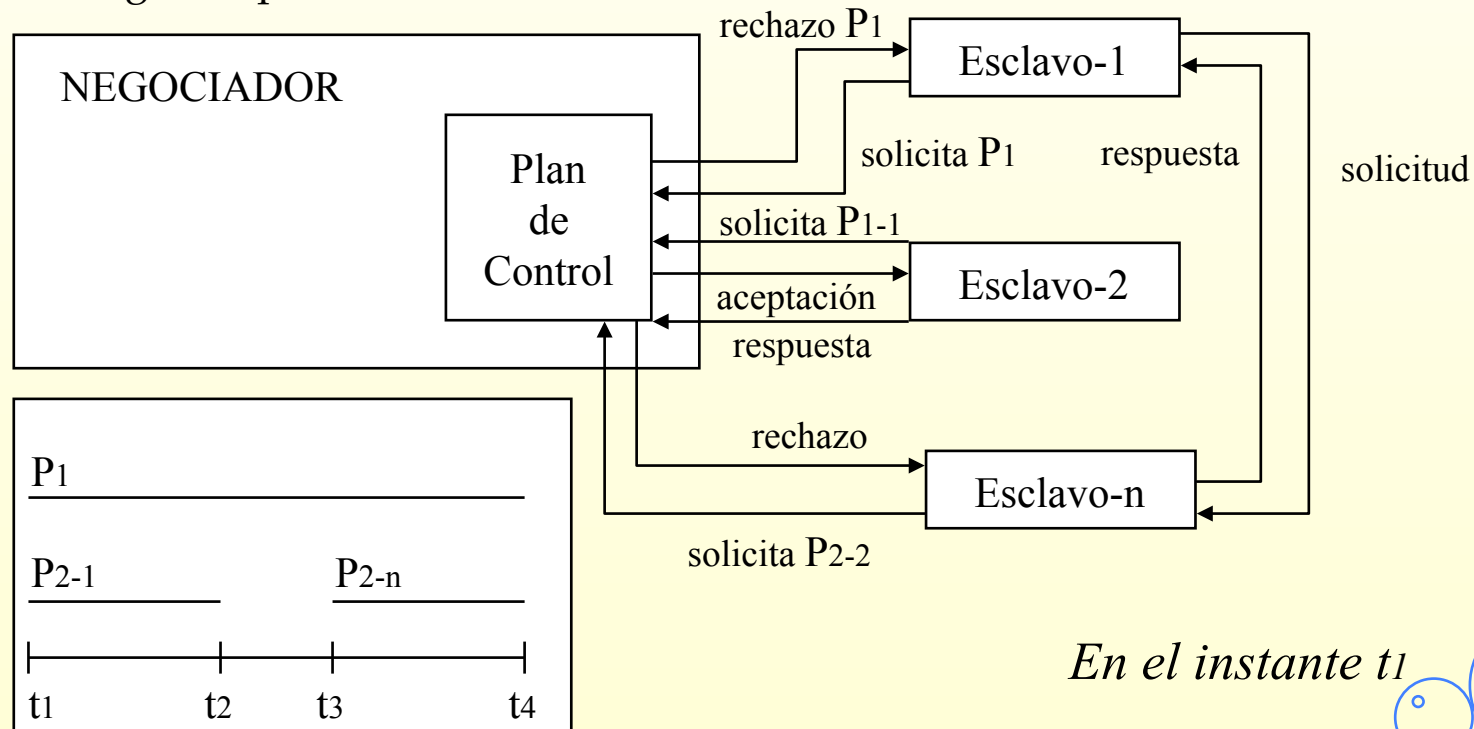
En el instante t1



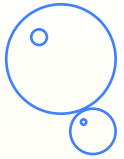
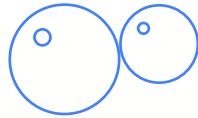
Coordinación asistida con negociador

- Los agentes que ejecutan los planes "negocian" con el negociador para decidir cuál de ellos realiza el subplan.

- El negociador decide, por ejemplo, en función del instante en el que se realiza la solicitud, del nivel de ocupación de los agentes, y de la organización jerárquica y temporal de los planes en el plan de control, cuál es el más capacitado.
- Los agentes pueden comunicar entre sí.



En el instante t1

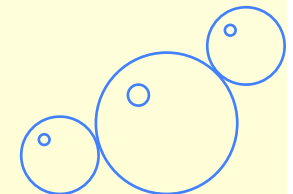
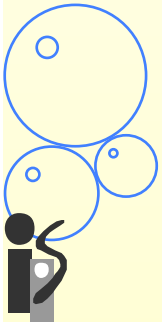


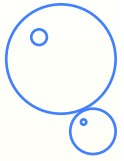
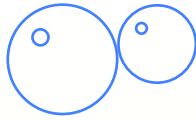
4.2 Con comunicación Directa

- **Ningún agente está encargado de realizar el plan**
 - Ventaja: No depende de la existencia, capacidades o influencias de otros programas
 - Inconveniente:
 - Elevado coste si el número de agentes es elevado.
 - Cada agente debe contener todo el código necesario para la negociación
 - Dos modelos de comunicación directa:
 - La aproximación de *red de contrato*

Los agentes que necesitan servicios distribuyen peticiones de propuestas a otros agentes. Los receptores evalúan y presentan ofertas a los emisores de peticiones.
 - La *compartición de especificaciones*

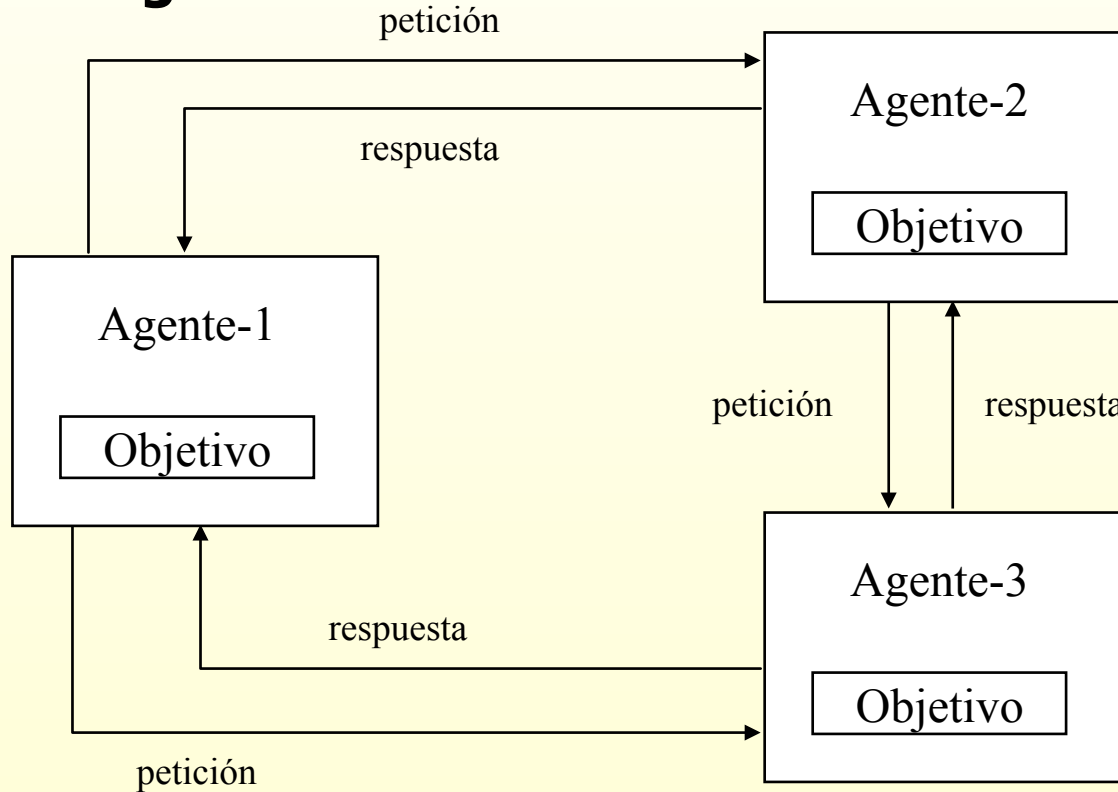
Los agentes suministran a otros agentes información sobre sus capacidades y necesidades. Estos agentes pueden usar esta información para coordinar sus actividades. Se disminuye enormemente la cantidad de información que fluye entre agentes.





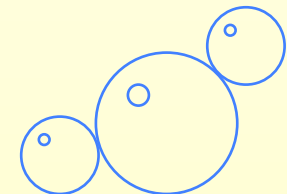
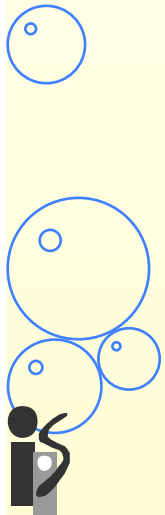
Comunicación directa cooperativa

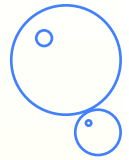
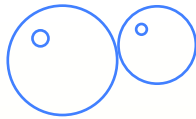
- Aunque ningún agente está encargado del plan, sí existe un objetivo común que es compartido por todos los agentes.**



El agente-1 solicita servicios de los agentes 2 y 3, los cuales envían sus ofertas al agente-1.

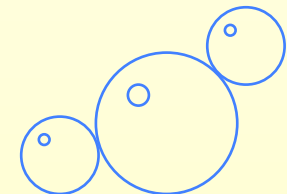
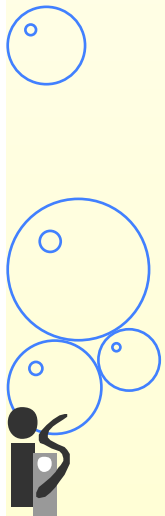
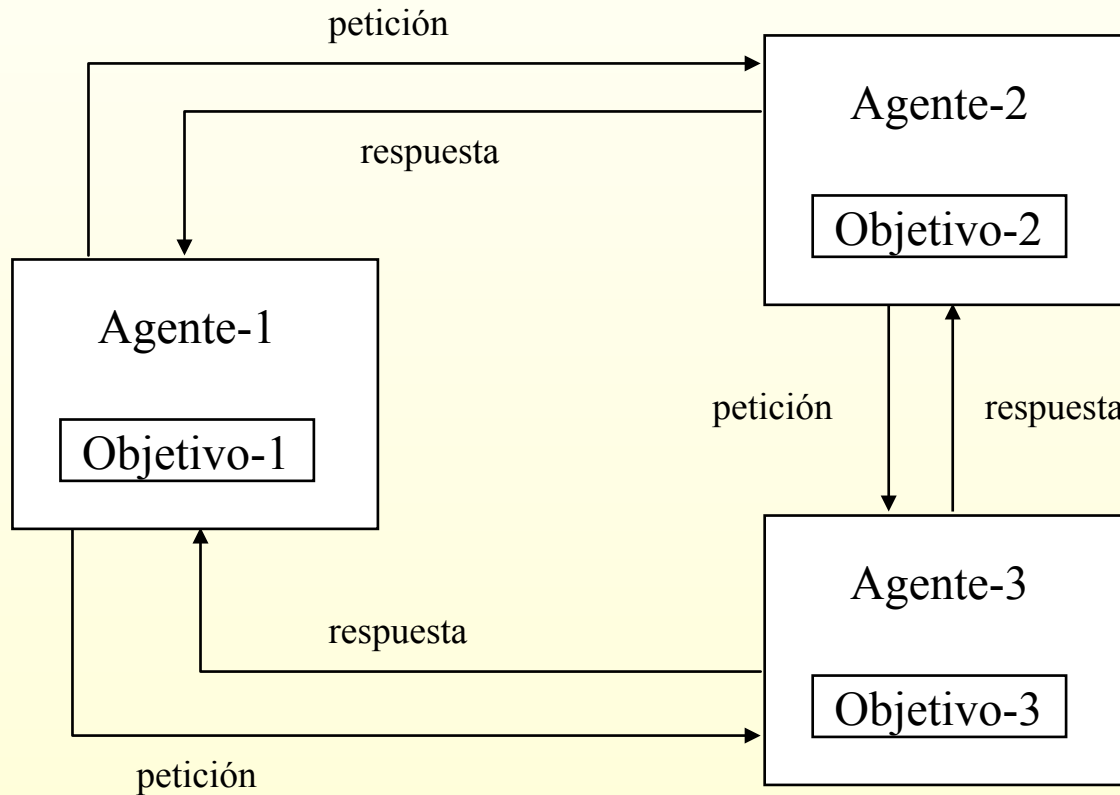
El agente 2 solicita sus servicios al agente-3

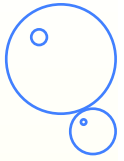
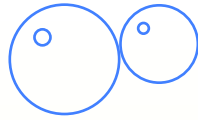




Comunicación directa anárquica

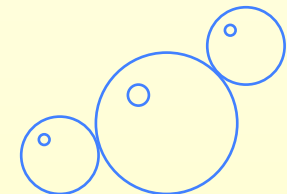
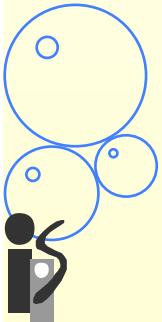
- **No existe un objetivo común compartido**

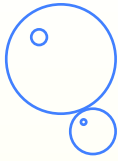
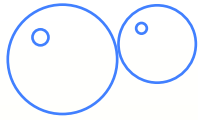




Técnicas de comunicación

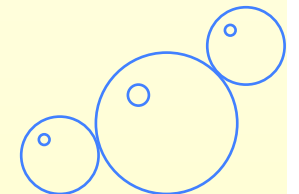
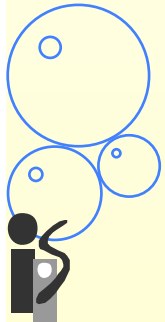
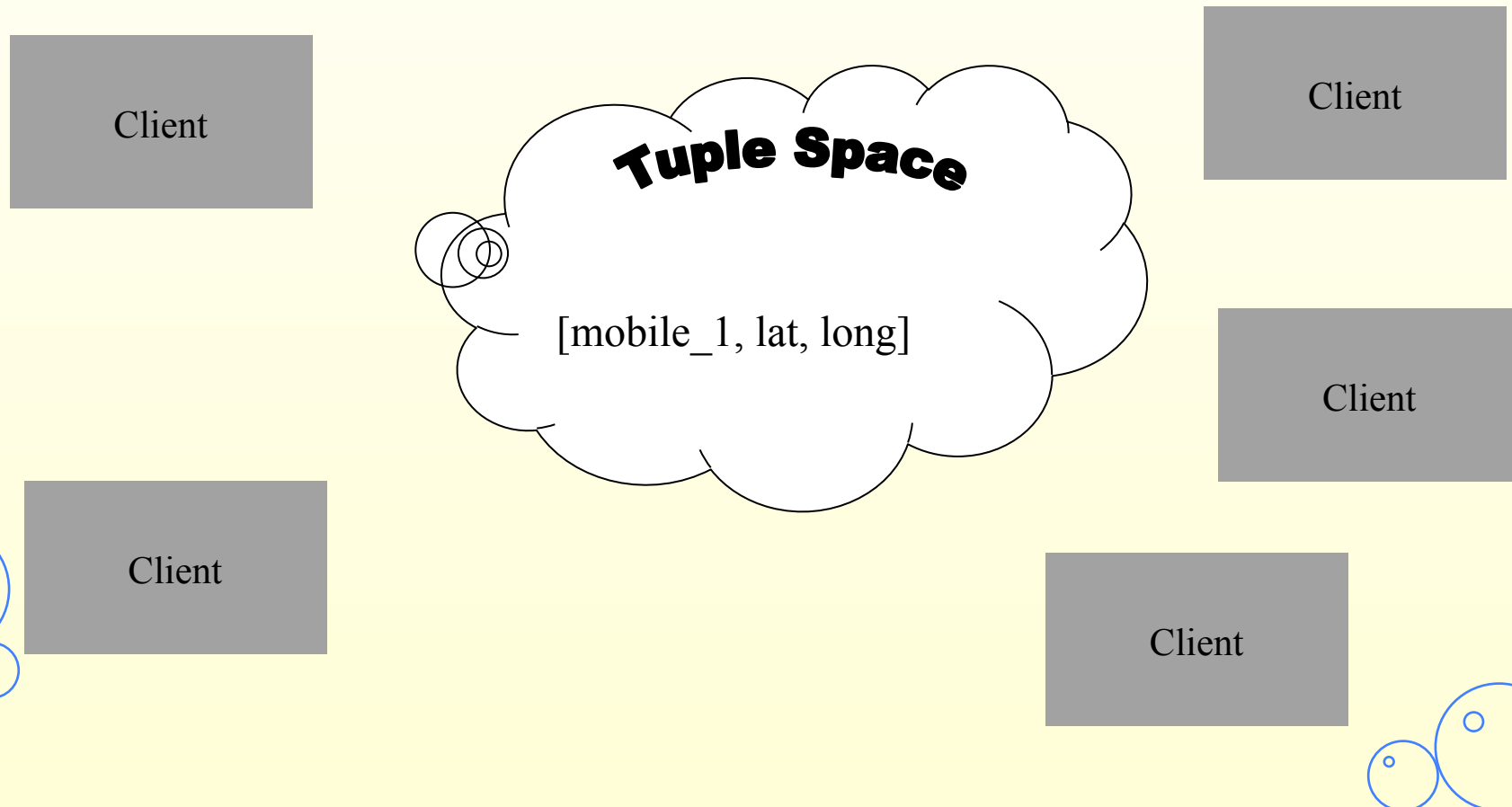
- **Existen diferentes técnicas para llevar a cabo la comunicación directa entre agentes**
 - Implementar en cada agente un módulo que soporte la negociación.
 - Cada agente conocerá los servicios ofertados por los otros (bien mediante un modelo de red de contrato o de compartición de especificaciones).
 - La **técnica de paso de mensajes** permite solicitar servicios conocidos de otros agentes.
 - Utilizar una estructuras de conocimientos compartida en la que se anotan mensajes
 - A los agentes les es imposible conocer los servicios ofertados por otros agentes.
 - La ejecución global consiste en una ejecución asíncrona de cada módulo

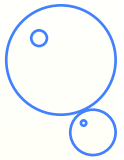
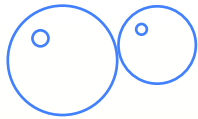




Modelo de Comunicación Generativa

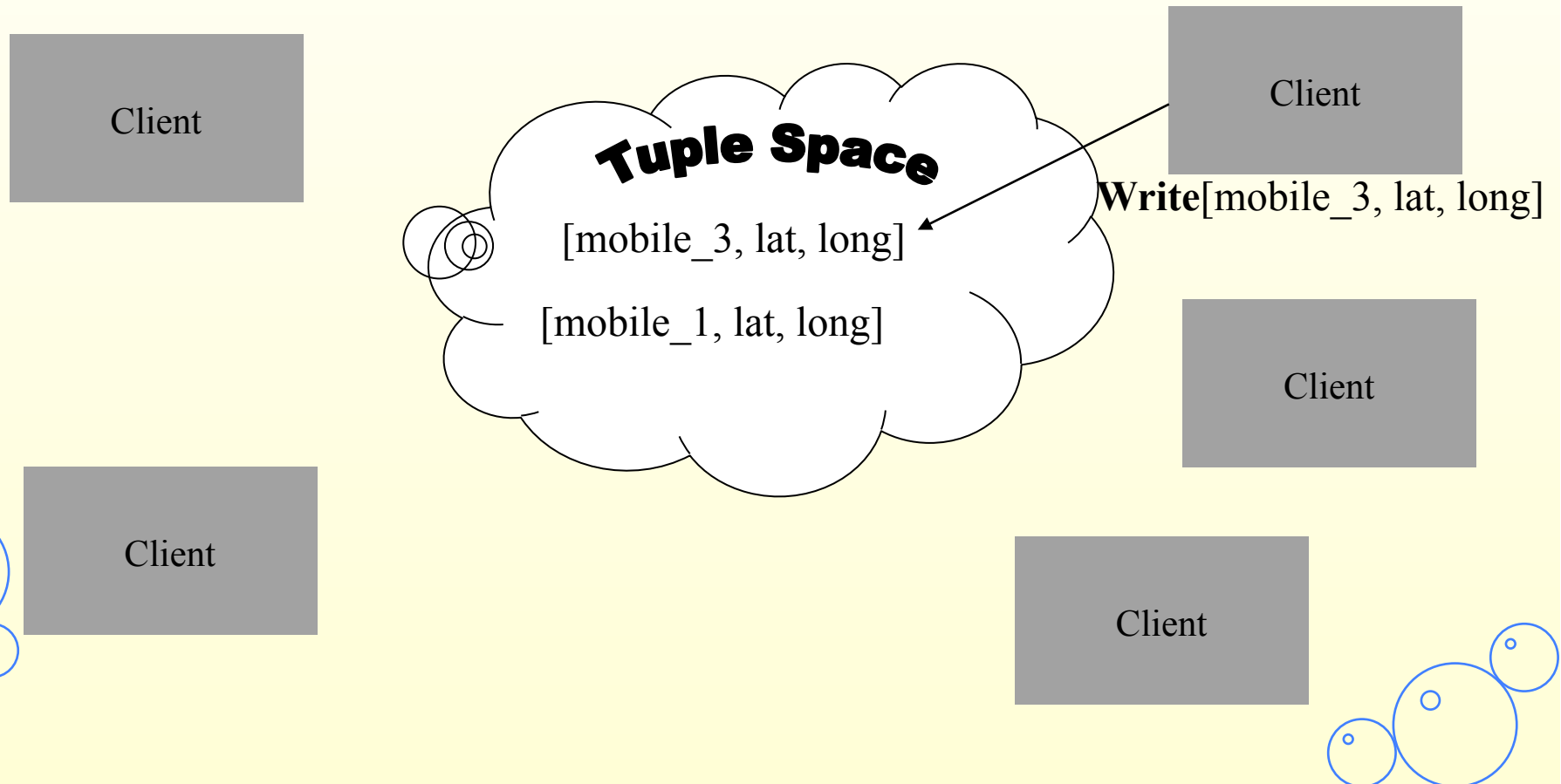
- **Generative Communication**

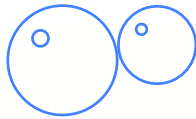




Modelo de Comunicación Generativa

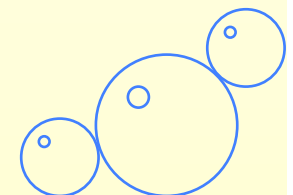
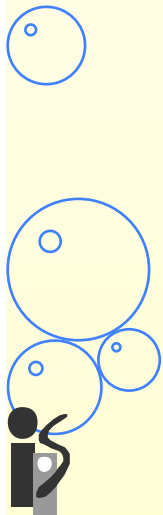
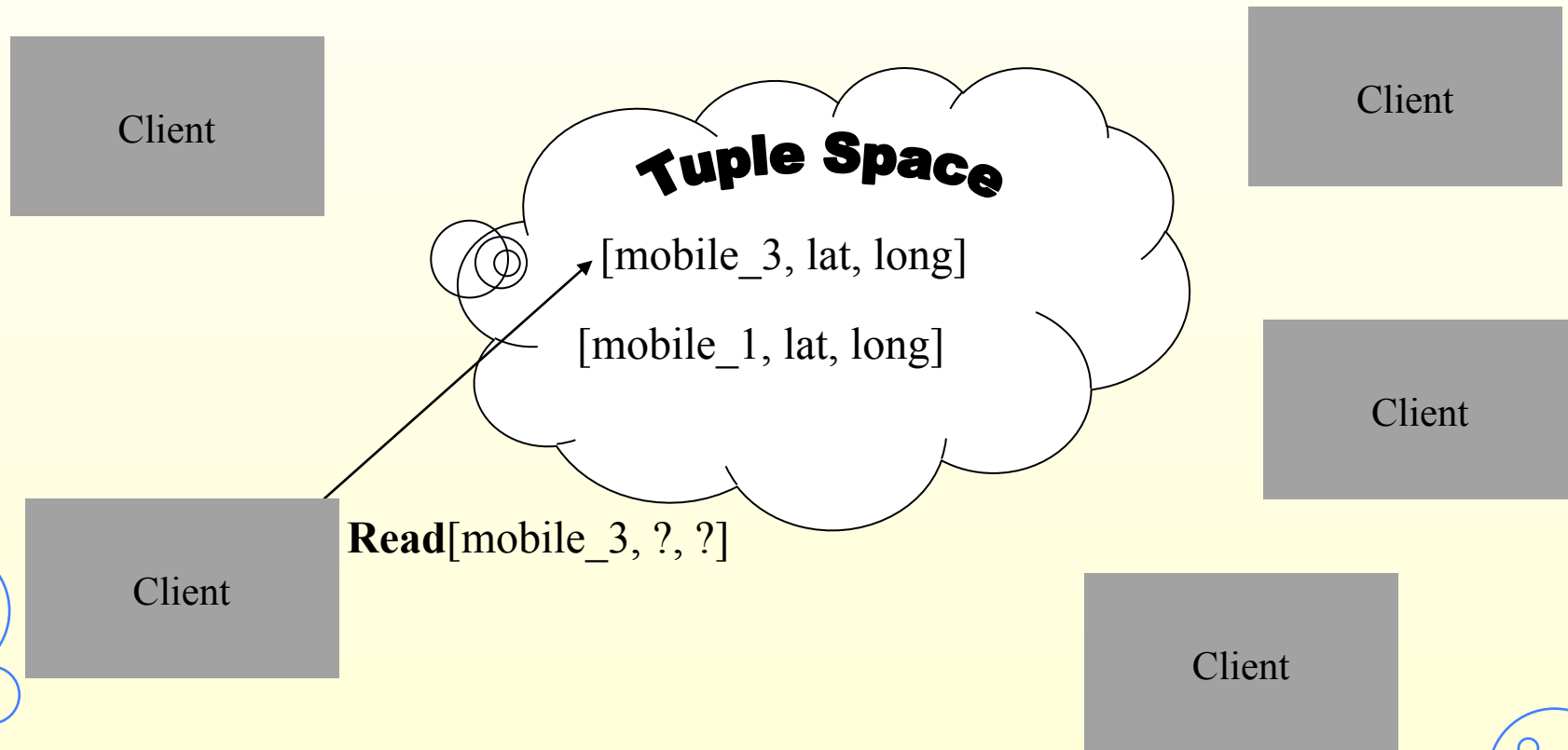
- **Generative Communication**

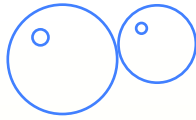




Modelo de Comunicación Generativa

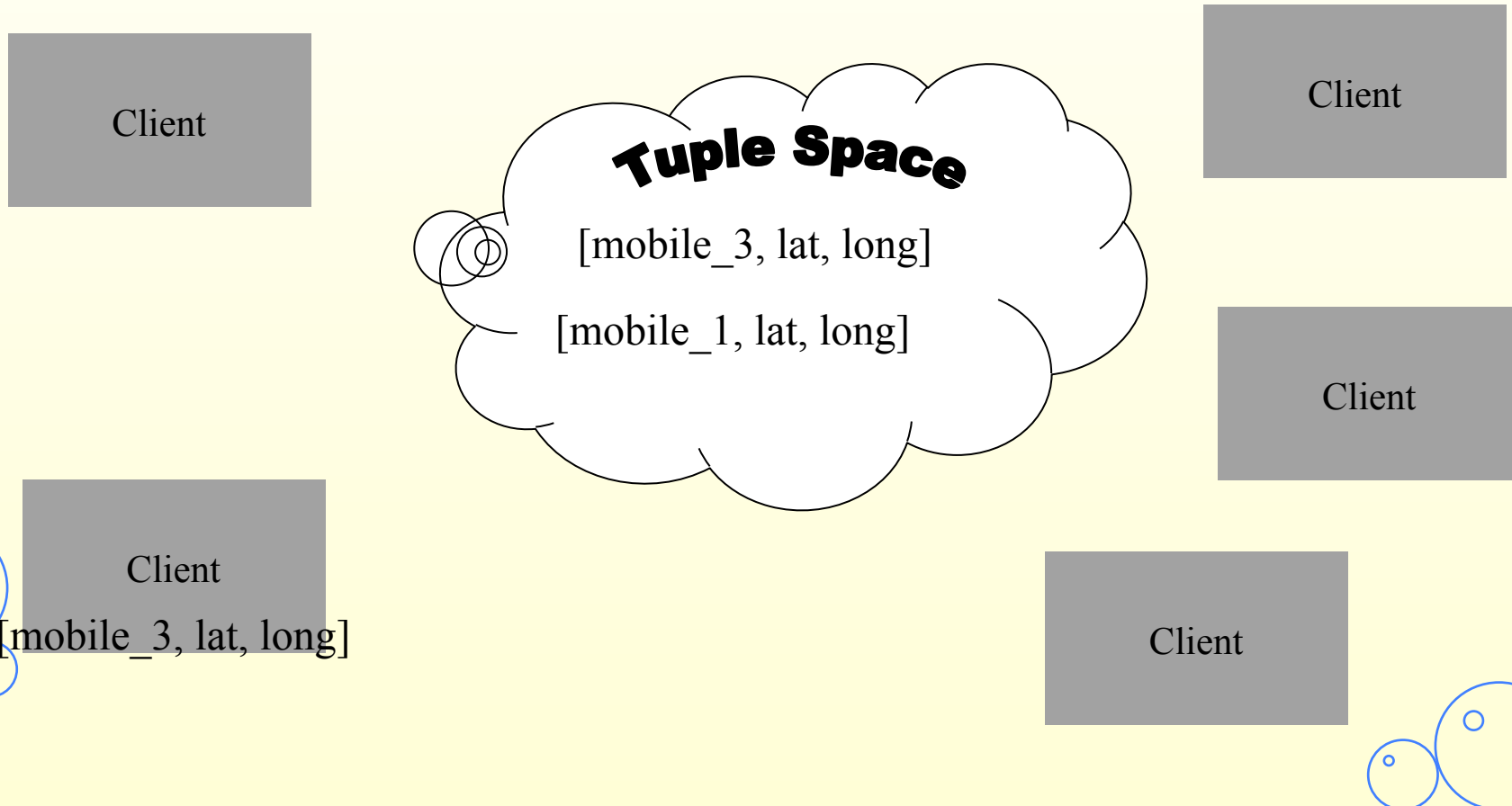
- **Generative Communication**

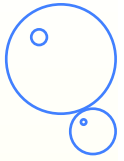
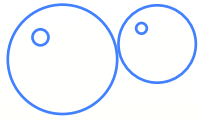




Modelo de Comunicación Generativa

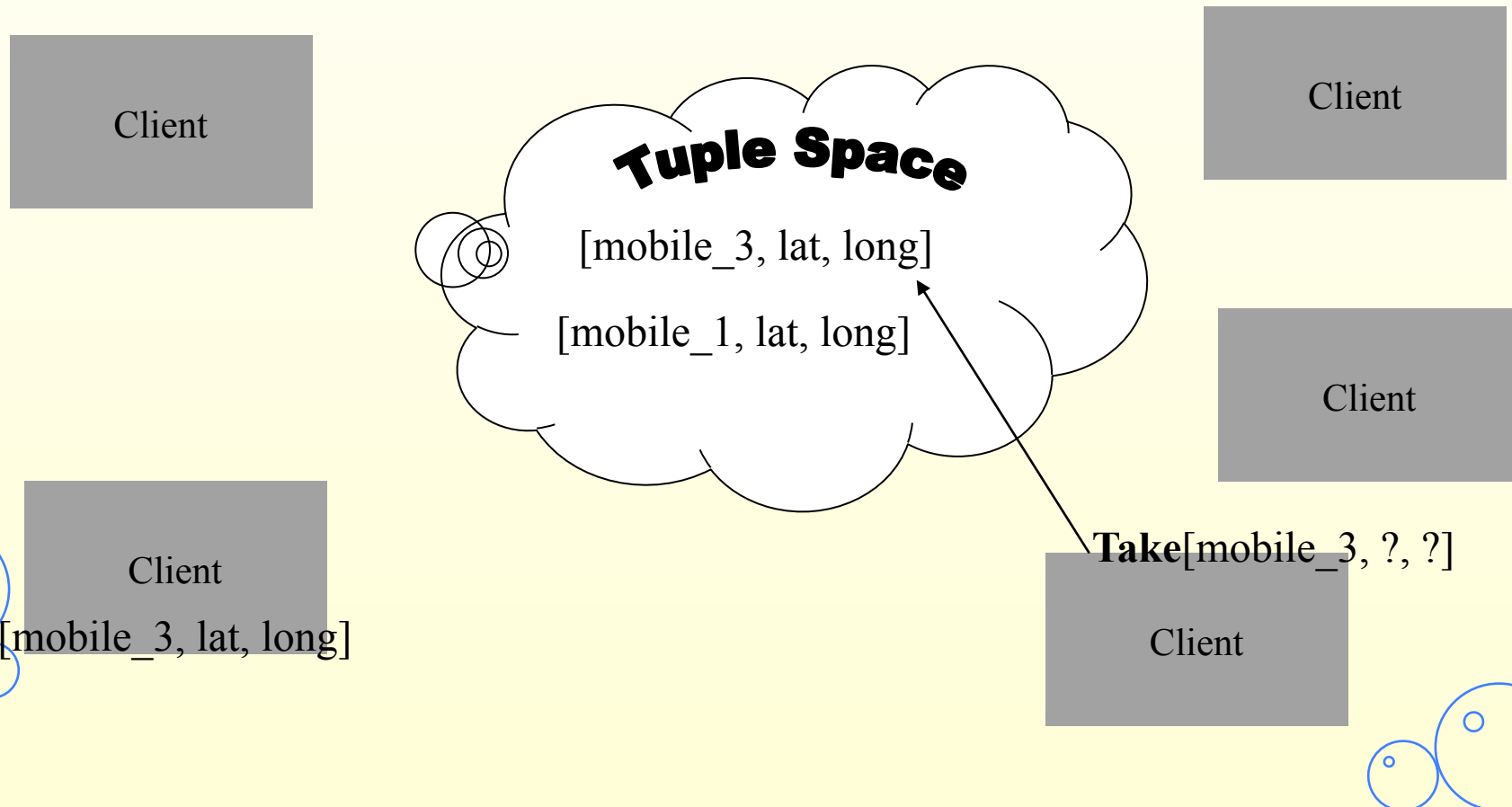
- **Generative Communication**

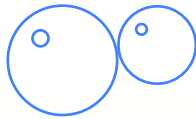




Modelo de Comunicación Generativa

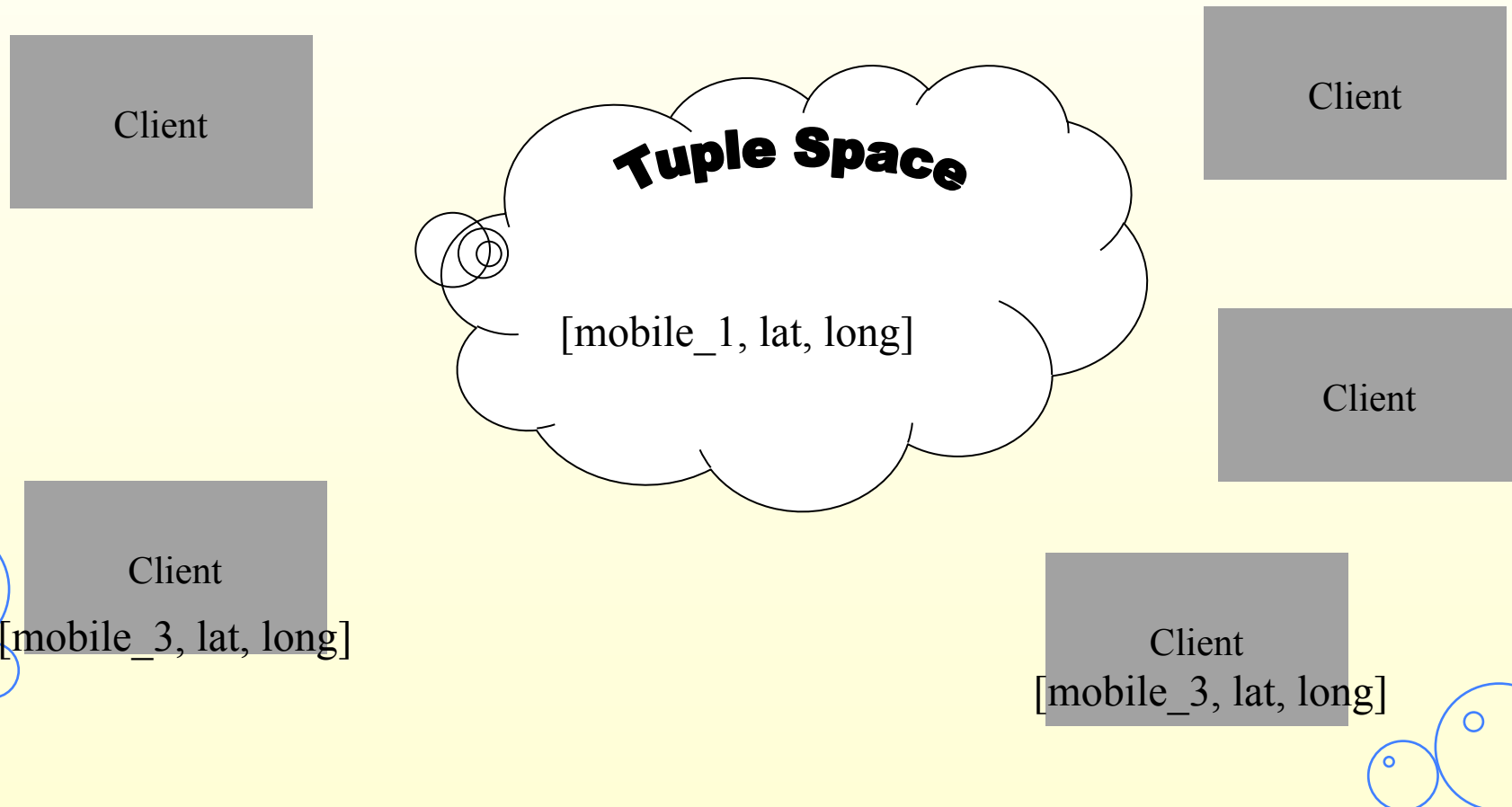
- **Generative Communication**

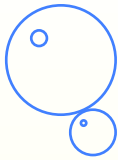
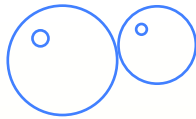




Modelo de Comunicación Generativa

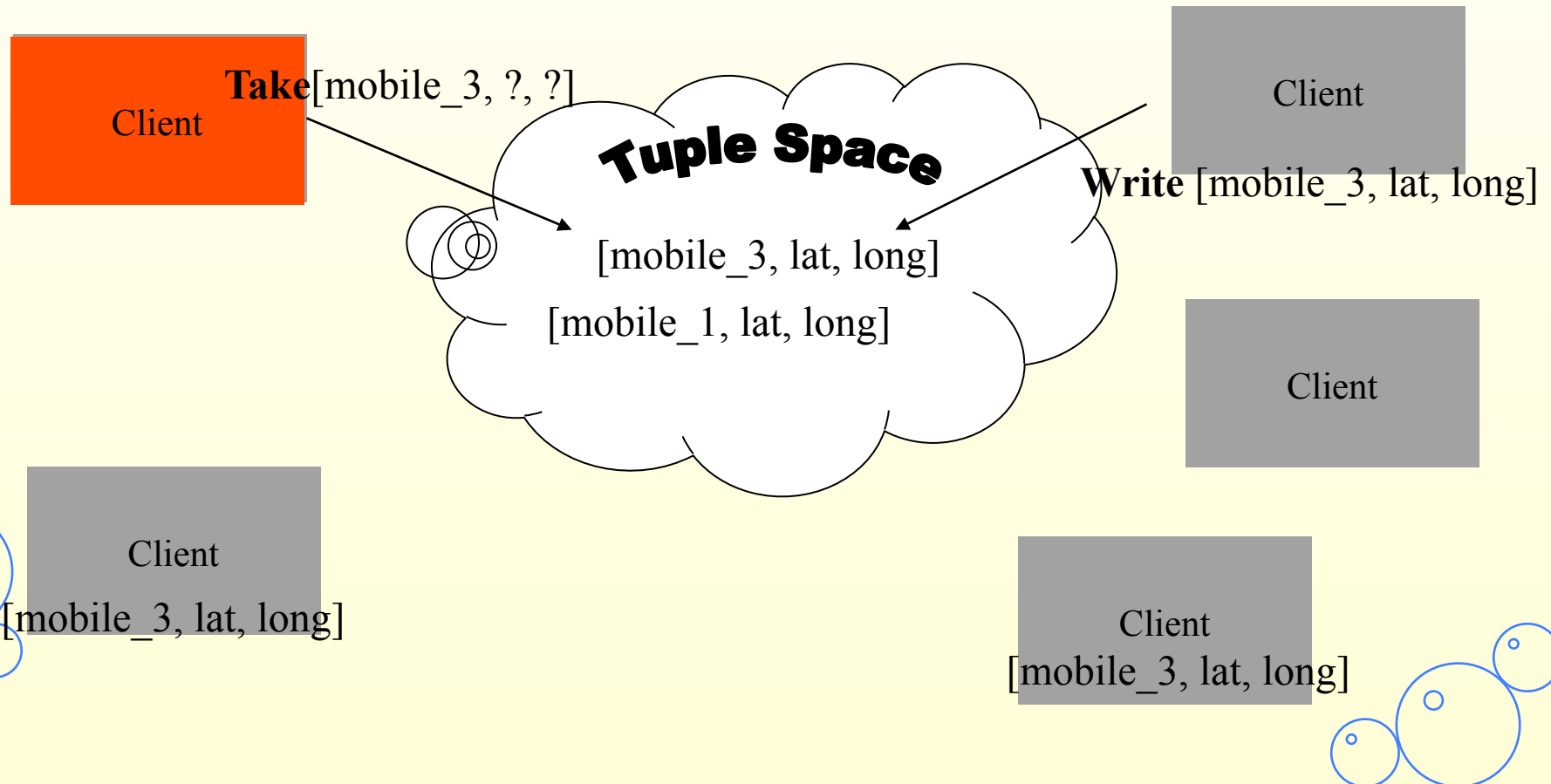
- **Generative Communication**

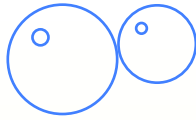




Modelo de Comunicación Generativa

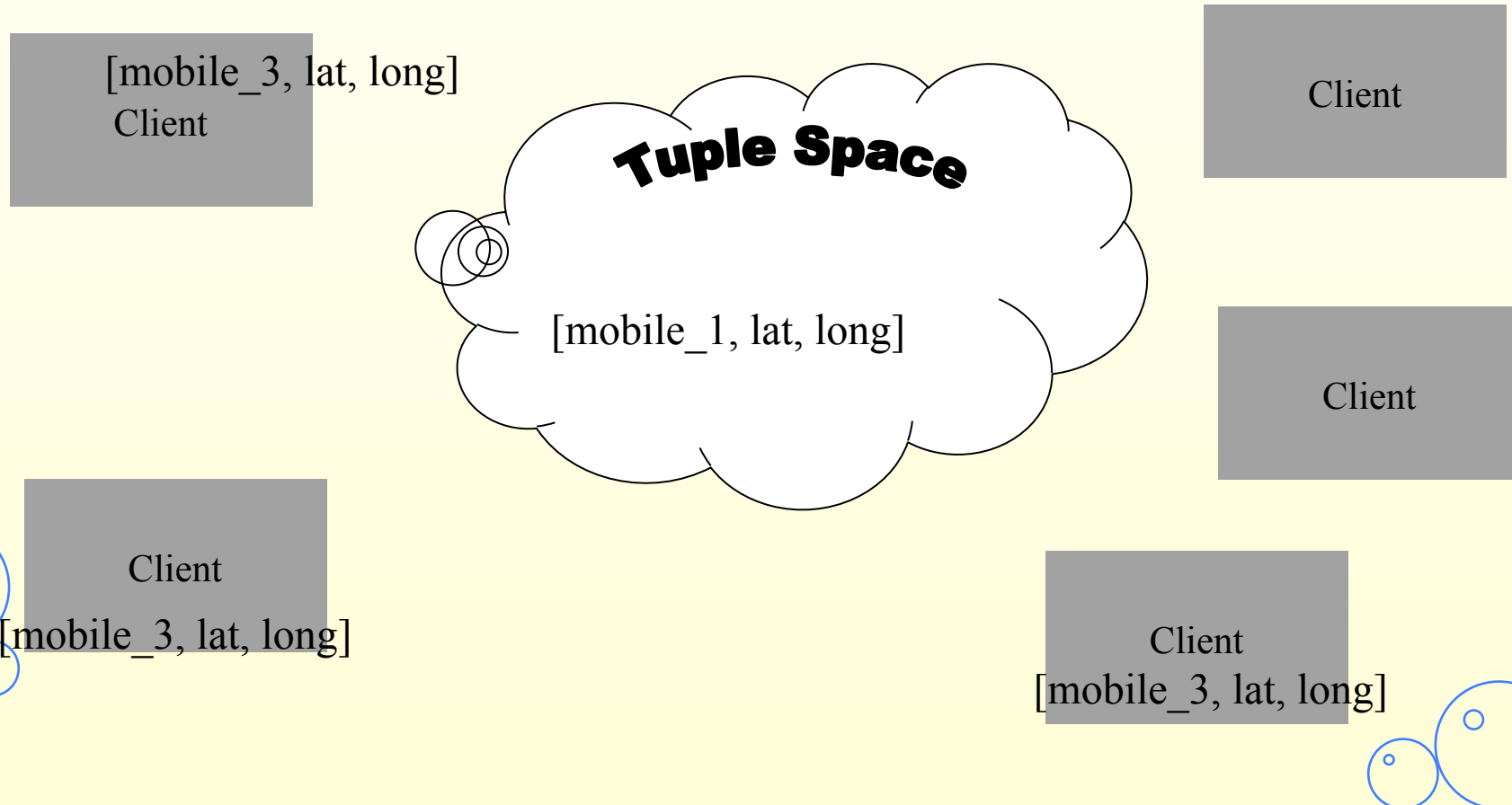
- **Generative Communication**

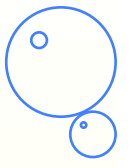
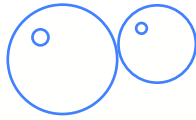




Modelo de Comunicación Generativa

- **Generative Communication**

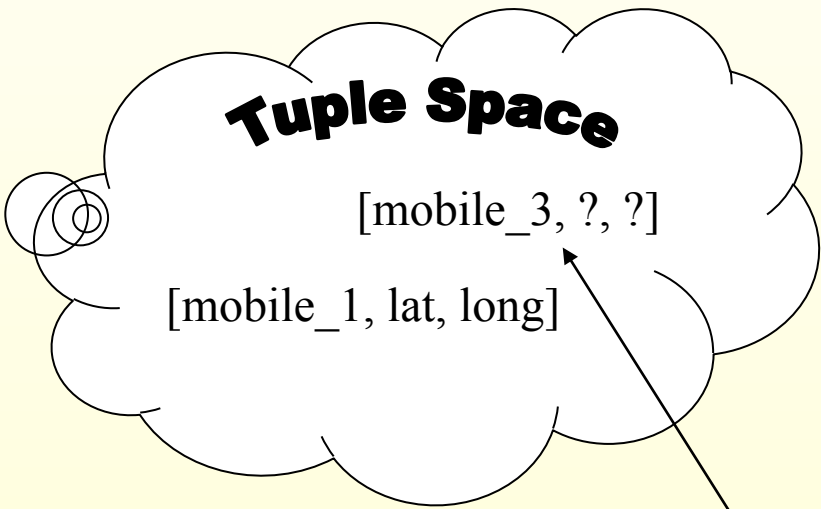




Comportamiento reactivo

Client

Client

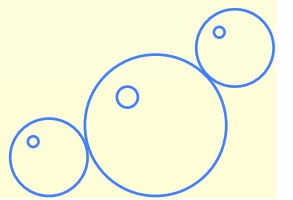
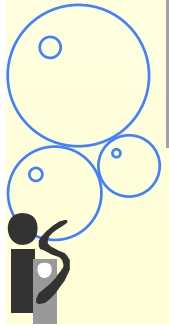


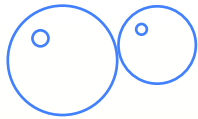
Client

Client

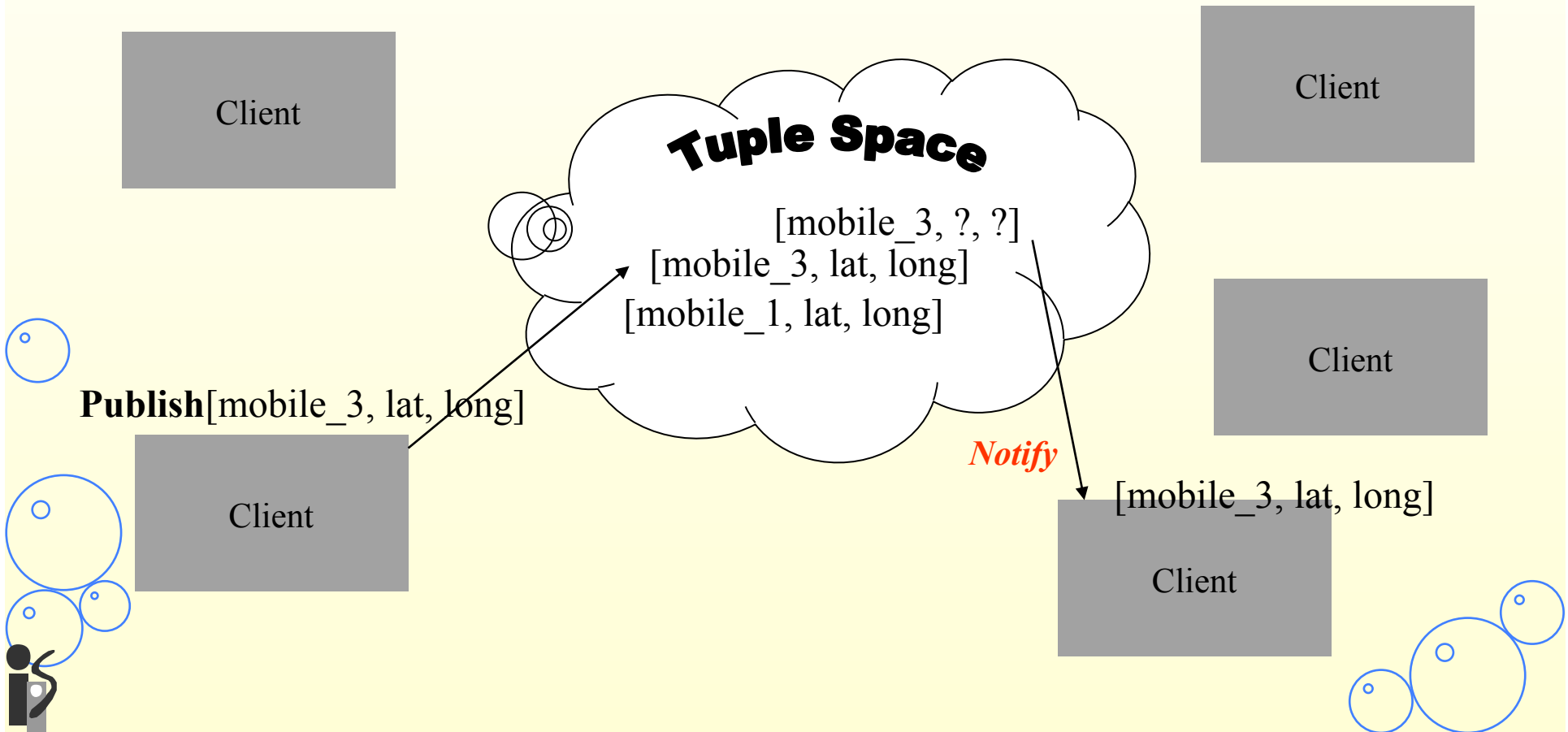
Subscribe[mobile_3, ?, ?]

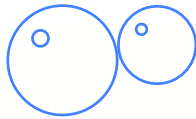
Client





Comportamiento reactivo





Comportamiento reactivo

Publish[mobile_3, lat, long]

Client

Tuple Space

[mobile_3, ?, ?]
[mobile_3, lat, long]
[mobile_3, lat, long]
[mobile_1, lat, long]

Client

Client

Client

Notify

[mobile_3, lat, long]

Client

[mobile_3, lat, long]

