



Conceptos y Arquitectura

Versión 6.0
Diciembre 2010

© KnowGate 2003-2010

© KnowGate 2010 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>
Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

Tabla de Contenidos

INTRODUCCIÓN.....	3
¿QUÉ ES HIPERGATE?	3
MÓDULOS FUNCIONALES INCLUIDOS EN EL PRODUCTO ESTÁNDAR.....	3
FUNCIONALIDADES COMUNES PARA TODOS LOS MÓDULOS	5
VENTAJAS Y BENEFICIOS DEL PRODUCTO	6
ARQUITECTURA	7
REQUISITOS Y PRIORIDADES DE DISEÑO.....	7
LENGUAJES, COMPONENTES Y PLATAFORMAS UTILIZADOS.....	10
JAVA Y TOMCAT	10
SISTEMAS GESTORES DE BASE DE DATOS RELACIONAL SOPORTADOS	11
SERVIDOR DE APLICACIONES	11
ESTRUCTURA INTERNA.....	11
DISEÑO MULTI-CAPA	11
SERVIDOR SIN ESTADOS	12
CACHES.....	12
SEPARACIÓN DE DATOS POR CLIENTES Y DEPARTAMENTOS.....	12
MODELO DE SEGURIDAD	14

1

Introducción

hipergate es una suite de aplicaciones basadas en web. La misión del producto es cubrir las necesidades esenciales de tecnología de información de las organizaciones modernas basándose en la disponibilidad de comunicaciones de banda ancha. hipergate posee capacidad multi-entidad y es apropiado para uso interno o en modalidad ASP¹.

¿Qué es hipergate?

hipergate es una suite de aplicaciones de código abierto basadas en web. El propósito del conjunto de programas es cubrir un amplio rango de necesidades de tecnología de información en cualquier organización. Todas las aplicaciones se acceden desde Internet Explorer sin necesidad de descargar ningún software en el puesto cliente.

La suite tiene capacidad multi-entidad y puede utilizarse indistintamente para dar servicio a una empresa, a un grupo de empresas, o en modalidad SaaS para alojar un número ilimitado de entidades cliente diferentes.



Módulos funcionales incluidos en el producto estándar

Modulo de Herramientas Colaborativas y Trabajo en Grupo

- Correo basado en Web.
- Calendario y Agenda de Reuniones Compartida.
- Foros Libres y Moderados con múltiples grupos.
- Área de Preguntas Frecuentes.
- Directorio de Personal
- Listado y Reserva de Salas y otros Recursos Compartidos

¹ SaaS. Software As A Service. Modalidad de servicio de aplicaciones alojadas en la que el proveedor alquila una máquina con todo el software necesario para su uso a través de Internet por múltiples clientes.

© KnowGate 2010 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>

Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

Modulo de Gestión de Contactos

- BB.DD. de Clientes, Proveedores, Competidores y Partners.
- BB.DD. de contactos personales
- Múltiples Direcciones por Contacto.
- Gestión de Demarcaciones Territoriales –Delegaciones-
- Gestión del Pipeline de Ventas -Oportunidades Comerciales
- Gestión de cuentas y ventas por comercial-
- Listas de Distribución de diversos tipos.
- Carga Directa de Windows Address Book -Outlook Express-.
- Carga Directa de ficheros de Contactos.

Modulo de Gestión de Proyectos y Soporte a Incidencias

- Árbol Jerárquico de Proyectos.
- Seguimiento de Tareas Pendientes.
- Control de Averías e Incidencias.
- Imputación de costes y horas a proyectos
- Gestión de partes de trabajo semanales
- Exportación de subproyectos y tareas a gráficos Gantt
- Contratos de Mantenimiento con Clientes.

Modulo de Tienda Virtual

- Múltiples Catálogos Independientes.
- Jerarquía ilimitada de Categorías de Productos.
- Atributos Variables por Producto.
- Gestión de Stock en múltiples almacenes.
- Gestión de Pedidos.
- Gestión de Facturación.
- Gestión de pagos online

Modulo de Producción de Contenidos

- Plantillas para comunicación via e-mail.
- Inclusión de contenidos multimedia.
- Gestión categorizada de contenidos.
- Librería de portlets para la presentación de contenidos dinámicos.

Modulo de Envío Masivo de Correos Electrónicos

- Gestión de envíos múltiples de e-mails a listas de distribución.
- Estadísticas detalladas de efectividad del email marketing.

Biblioteca Corporativa

- Disco Virtual 100% basado en Web.
- Control de versiones.
- Seguridad por usuario basada en roles para los archivos.
- Gestión e Indexación de propiedades de documentos OLE.
- Enlaces Favoritos compartidos.

Marketing

- Definición y seguimiento de campañas.
- Convocatoria, promoción y gestión de asistencia a eventos.
- Espía de conversaciones en la web, Twitter, Facebook, LinkedIn, etc.

Funcionalidades Comunes para todos los módulos

Seguridad basada en roles

- Múltiples dominios independientes de seguridad.
- Número ilimitado de usuarios y grupos de permisos.
- Roles predefinidos listos para utilizar en las aplicaciones.
- Seguridad a nivel de empresa y a nivel de departamento.
- Integración con LDAP.

Gestión de datos jerárquicos

- Categorización de objetos.
- Tesauro geográfico.
- Tesauro lexicográfico.

Informes y Consultas

- Herramienta de Consulta Guiada por Formulario.
- Volcados a Excel.
- Consultas online en formato HTML.
- Informes remitidos por e-mail.

Planificador de Tareas

- Gestión de colas de procesos.
- Ejecutor multi-hilo en segundo plano.

Ventajas y Beneficios del Producto

hipergate es software de última generación; diseñado, programado y testeado con las herramientas y técnicas más modernas.

Aplicaciones independientes pero sinérgicas.

Cada aplicación de la suite puede funcionar de forma independiente o junto con las otras aplicaciones. Las aplicaciones pueden activarse o desactivarse en caliente de forma particular para cada cliente, empresa o departamento. Esto hace que el usuario final pueda contratar las aplicaciones que realmente necesita sin tener que comprar toda la solución como sucede con los paquetes de software actuales.

Acceso seguro y unificado.

El usuario sólo tiene que autenticarse una sola vez para todas las aplicaciones.

Integración con Office.

hipergate reconoce e indexa documentos Office, permite cargar y exportar datos a Outlook Express y generar consultas en formato Excel.

Solución Integral.

hipergate está concebido y pensado tanto en profesionales independientes como en empresas, de manera que puedan derivar todos sus procesos informáticos hacia la plataforma. La suite está orientada a facilitar el trabajo diario, creando así una base de usuarios que utilicen el software constantemente.

Registro rápido y a trabajar.

Al usuario se le da opción de que pruebe la herramienta durante un periodo de tiempo. El usuario puede en cualquier momento contratar el uso de las aplicaciones hipergate mediante pago a través de Internet aunque también se dan opciones de otras formas de pago. Una vez realizado el pago el usuario puede empezar a trabajar con las aplicaciones contratadas.

Entorno intuitivo y fácil de manejar.

Cada aplicación es fácilmente accesible desde un menú principal de pestañas. El interface con el usuario está diseñado para ser homogéneo y de esta forma de fácil y rápido aprendizaje.

Auto-administrable por el usuario final.

hipergate suministra un modelo de administración predefinido basado en 4 perfiles de usuario: *administrador*, *superusuario*, *usuario* e *invitado*; a los que se asocia unas operaciones que pueden hacer sobre su información de forma muy parecida a como funciona el sistema operativo.

El administrador de la instancia concreta.

Las facilidades de auto-administración sirven dos propósitos: 1º) reducir el coste de propiedad de la aplicación en el cliente y 2º) reducir el coste de mantenimiento si la plataforma se ofrece en modalidad ASP.

Arquitectura

2

Un diseño grande y complejo, que no ha evolucionado a partir de otro pequeño y muy bien pensado, no funciona, y, además, es imposible arreglarlo a posteriori para que funcione.

Bjarne Stroustrup,
The Design and Evolution of C++

Requisitos y Prioridades de Diseño

hipergate ha sido diseñado desde el principio como una suite orientada a dar un servicio de alta calidad a los clientes más exigentes.

Se han fijado una serie de requisitos y prioridades que el producto debe cumplir estrictamente.

Ergonomía y Usabilidad.

La primera prioridad del producto es que cada parte proporcione una buena experiencia de usuario. Esto se consigue mediante:

- arquitectura de información homogénea a través de todos los módulos.

© KnowGate 2010 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>

Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

- modelo conceptual sencillo.
- fácil acceso a todas las funcionalidades.
- tiempos de respuesta rápidos para todas las operaciones.
- completitud en la gama de operaciones disponibles.

Alcance Funcional.

La suite está pensada para cubrir el 80% de las necesidades más típicas de cada módulo funcional. En este apartado se prefirió la expansión horizontal del producto por múltiples módulos antes que fabricar unas pocas aplicaciones específicas de alta complejidad.

La filosofía del producto es que las empresas pequeños y medianas obtengan prácticamente todas las funcionalidades necesarias para cada departamento y que las grandes dispongan de una base sólida para desarrollar extensiones propietarias.

Estabilidad.

La suite es un producto pensado para tener disponibilidad 24x7. Cada nuevo módulo es meticulosamente testado en sucesivos *walk throughs* de código, pruebas de caja negra y pruebas de stress en condiciones críticas.

Escalabilidad.

Durante todo el ciclo de desarrollo no se quiso sacrificar en ningún momento el rendimiento por la transportabilidad o la ampliación funcional del producto. El código está optimizado para hacer uso de las funcionalidades específicas de cada base de datos y plataforma sobre los que puede ejecutarse.

Una parte de la lógica de proceso emplea procedimientos almacenados PL/SQL, PL/pgSQL o Transact-SQL reescritos manualmente para aprovechar al máximo las opciones más avanzadas de cada SGBDR.

El código Java dispone de 3 modos de ejecución: 100% Java Puro, Unix y Win32, y, según se configure explota las llamadas atómicas de cada sistema operativo para obtener el mejor rendimiento.

La aplicación está diseñada en 3 capas: servidor web, servidor de aplicación y servidor de base de datos. El diseño hace énfasis en la capacidad para crear granjas y repartir la carga entre múltiples servidores.

Para reducir el consumo de memoria y aumentar la capacidad de servicio de cada servidor web, la aplicación funciona sin sesiones ni estados mantenidos en el lado servidor.

Por último, la suite incorpora un sofisticado sistema propietario de cache distribuido con la misión de mantener información local en los servidores web y reducir la sobrecarga de los nodos de la base de datos.

Tolerancia a Fallos.

La aplicación puede configurarse para funcionar en clustering tanto con los servidores web como con la base de datos.

Mantenibilidad.

La estructura del código se diseñó pensando en que resultase sencillo de mantener y de ampliar por programadores sin conocimientos profundos de los detalles del sistema. El modelo de objetos que aísla el modelo físico de la base de datos del modelo de la lógica de negocio proporciona un marco natural para hacer ampliaciones con una curva de aprendizaje suave. Muchas de las tareas rutinarias de codificación: generación de formularios, tablas de remonte, validación de datos, gestión de fechas, etc. ya están resueltas de forma estándar en componentes reutilizables.

Separación de datos entre clientes.

hipergate comparte información de varios clientes en una misma base de datos para evitar una proliferación insostenible de bases de datos. No obstante, como requisito imprescindible para poder dar servicio ASP, la información de cada cliente puede extraerse limpiamente en cualquier momento de la base de datos compartida para hacer una copia de respaldo exclusiva del cliente o para instalarse en una instancia de base de datos dedicada.

Economía.

La aplicación puede funcionar basada 100% en software libre con el propósito de eliminar casi por completo los costes de licencias. Otro factor, es el uso racional de la CPU y el disco que se consideran recursos escasos.

Tecnología Estándar.

Sólo se emplean componentes y tecnologías de uso muy común en el mercado. Además se aboga por el uso exclusivo de plataformas con respaldo explícito para su continuidad de las grandes empresas del sector.

Simplicidad.

A pesar de su amplio alcance técnico y funcional, la suite está diseñada y codificada para ser muy simple parte por parte.

Lenguajes, Componentes y Plataformas Utilizados

Java y Tomcat

Todos los módulos de hipergate están escritos en 100% Java Puro.

La versión 6.0 de hipergate ha sido testeada sobre Tomcat 6.0 con Java 1.6.

La máquina que aloje el servidor web debe ser Windows, Linux u OpenSolaris.

Actualmente no se soporta Tomcat sobre BSD. La suite no tiene ninguna restricción específica sobre estas plataformas, pero la instalación del servidor web sobre BSD.

Componentes utilizados bajo licencias Open Source

- Jakarta Bean Scripting Framework 2.3
- Jakarta POI 2.5
- Xerces2 XML Java Parser 2.9
- Xalan XSLT Processor 2.7.1
- Pat Niemeyer BeanShell 2.0
- Enterprise DT Ltd Java FTP Library 1.2.2
- Jakarta ORO 2.0.8
- Jamie Jaworski DHTML Tabbed Panel
- DynAPI Cross-Browser DHTML Library

Componentes utilizados bajo licencia de SUN Microsystems Inc.

- J2SE 1.6
- JavaBeans™ Activation Framework 1.0.2
- JavaMail™ 1.4
- Java™ Advanced Imaging 1.1.2

Otros Componentes

- Infomentum AppletFile 3.0 (opcional)

Sistemas Gestores de Base de Datos Relacional Soportados

- PostgreSQL 9.0
- Microsoft SQL Server 2008
- Oracle 10g
- MySQL 5.x

Servidor de Aplicaciones

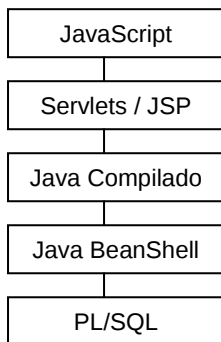
El la persistencia objetos en base de datos, el módulo de cache distribuido y el planificador de tareas pueden instalarse separados del servidor web sobre un servidor de aplicaciones J2EE.

Estructura Interna

Diseño multi-capa

El código de hipergate está dividido en 5 capas:

Capas



- **Capa 1:** Código JavaScript ejecutado por el navegador cliente.
- **Capa 2:** Páginas JSP servidas por el servlet runner (Tomcat)
- **Capa 3:** Modelo abstracto de objetos Java, puede ejecutarse dentro de Tomcat o de forma separada sobre un servidor J2EE.
- **Capa 4:** Scripts Java BeanShell.
- **Capa 5:** Procedimientos almacenados en el SGBDR.

Esta división persigue alcanzar el mejor grado de escalabilidad y extensibilidad de la aplicación, con los objetivos de:

1. Realizar el máximo de validaciones y procesamiento en cliente para reducir el tráfico entre el navegador y el servidor web.
2. Separar la capa de presentación de la persistencia de objetos.
3. Proporcionar un API para todos los objetos de la aplicación.
4. Utilizar el máximo de código compilado y optimizado para el núcleo de librerías del sistema.
5. Externalizar la lógica de negocio en scripts de servidor que no requieran ser recompilados para ejecutarse.

© KnowGate 2010 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>

Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

6. Reducir el número de llamadas a la base de datos en aquellas operaciones que puedan ejecutarse de forma atómica dentro del gestor de base de datos.

Servidor sin estados

hipergate no utiliza sesiones ni estados mantenidos en el servidor.

Esto es una medida destinada a reducir el consumo de memoria y aumentar la escalabilidad en el servidor web.

Toda la información se mantiene mediante cookies² de sesión almacenadas en el cliente.

Estas cookies sólo contienen un mínimo de información:

- dominio de seguridad al que se está conectado
- área de trabajo a la que se está conectado
- identificador único de del usuario conectado
- token de acceso de sesión encriptado

Como no hay sesiones, no existe el concepto de identificador de la sesión actual. Toda la información de estado se transfiere de una página a otra mediante métodos GET o POST de HTTP.

Caches

El sistema utiliza un cache distribuido que permite almacenar información de la base de datos de forma local en los servidores web. Esto reduce el tráfico de red y aligera la carga sobre la base de datos.

Un controlador de cache se encarga de mantener la coherencia de los datos en instalación con múltiples servidores web funcionando de forma concurrente contra la misma base de datos.

Separación de datos por clientes y departamentos

² Los cookies son pequeños archivos de texto almacenados en el ordenador cliente de la aplicación. Estos archivos también se denominan en ocasiones *huellas*. Sirven para mantener información entre llamadas HTTP para un mismo cliente.

© KnowGate 2010 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>

Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

Muchas aplicaciones que funcionan en modalidad ASP lo hacen replicando automáticamente un modelo de datos para cada una de las instancias cliente. La ventaja de este enfoque es que permite separar fácil y limpiamente los datos de cada cliente. El inconveniente es que se genera una proliferación de bases de datos clónicas prácticamente imposibles de mantener a partir de cierto volumen de instancias.

En hipergate se ha seguido una aproximación híbrida al problema de la separación de datos: una única base de datos puede contener información de múltiples entidades cliente sin solapamientos, incluso, yendo un paso más lejos, la división de datos y permisos de acceso puede llegar a nivel departamental, consiguiendo que las personas de un departamento tengan acceso a un conjunto de datos y aplicaciones diferente de los de otro departamento.

Sin embargo, para mantener el requisito indispensable de la separación de datos por cliente, las utilidades de administración disponen de subrutinas para cortar y separar la información de un cliente en una base de datos aislada de uso exclusivo, incluso aunque dicha información se encontrase previamente almacenada en una base de datos compartida.

Dominios

Conceptualmente, el Dominio de hipergate es la unidad de mayor nivel de división de datos. Típicamente representa una entidad cliente completa, aunque otro uso común es usarlo como contenedor de usuarios individuales no adscritos a ninguna entidad (por ejemplo, profesionales independientes que contratan una cuenta de uso personal).

El uso principal de los dominios es establecer los límites de asignación de permisos para cada administrador. Así el administrador de una entidad cliente, puede crear nuevos usuarios o activar y desactivar aplicaciones dentro de su dominio, pero no puede ver ni alterar los datos de otras entidades clientes en dominios diferentes.

Áreas de Trabajo

Cada Dominio puede contener una o más Áreas de Trabajo. Las áreas de trabajo forman los compartimentos estancos de información.

Las áreas de trabajo suelen representar departamentos funcionales en las entidades cliente.

En un momento dado cada usuario sólo ve la información de aquella área de trabajo a la que está conectado con un rol determinado.

Por ejemplo, un vendedor podría ser usuario del área de trabajo del departamento de ventas, y, al mismo tiempo invitado del área de trabajo del departamento de soporte técnico. Este vendedor podría crear nuevas fichas de clientes o generar oportunidades comerciales en el área de trabajo de ventas, pero su nivel de privilegio en el área de trabajo de soporte sólo le permitiría ver las incidencias abiertas pendientes de resolución, sin posibilidad de modificarlas.

Modelo de Seguridad

hipergate implementa un modelo de seguridad a nivel de sesión basado en roles. El modelo maneja los siguientes conceptos :

Aplicación El producto funciona como un conjunto de aplicaciones interoperables. Se pueden añadir o quitar aplicaciones sin afectar al funcionamiento del resto del sistema. En un momento dado, cada usuario tiene disponible un subconjunto de todas las aplicaciones y ejerce un rol en cada una de ellas en función de los grupos de permisos a los que pertenezca.

Usuario Se permite crear un número ilimitado de usuarios cada uno con su propio conjunto de roles y clave de acceso.

Rol Se dispone de cuatro roles predefinidos más un quinto definible por entidad cliente: administrador, superusuario, usuario, invitado y variable. Estos roles marcan lo que el usuario puede hacer en cada aplicación. Así, un usuario podría tener rol de Administrador en el módulo del Directorio de Personal y Rol de Invitado en el módulo de ventas. Esto le permitiría crear nuevas fichas de empleados, pero no crear nuevos clientes.

Cuenta Las cuentas sólo se utilizan cuando el producto se instala en modalidad ASP. Una cuenta es equivalente a un contrato de uso con un cliente final.

Existen 3 tipos de cuenta predefinidas:

- cuenta corporativa
- cuenta profesional
- cuenta de sistema.

Las cuentas corporativas permiten un número arbitrario de usuarios y están asociadas a su propio dominio.

Las cuentas profesionales representan usuarios individuales que no pueden hacer uso de las funcionalidades de trabajo en equipo de la aplicación.

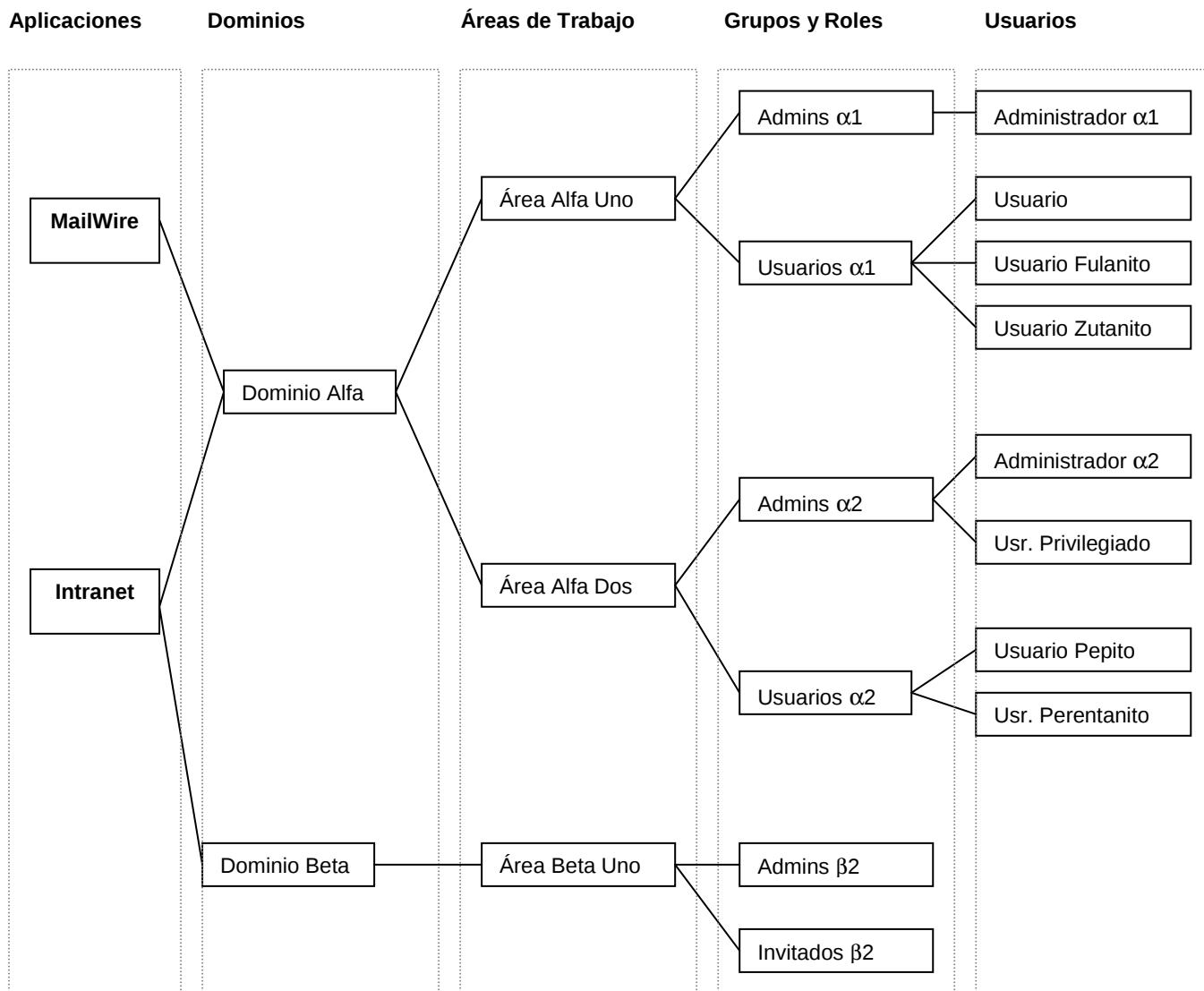
Las cuentas del sistema se usan por los operadores del ISP que proporcionan el servicio como un medio de administrar todos los dominios contratados con clientes.

Dominio Un dominio representa un conjunto de usuarios que se administran de forma independiente. Cada dominio tiene uno o más administradores.

Área de Trabajo Cada dominio contiene una o varias áreas de trabajo. El área de trabajo es una entidad que sirve para establecer la visibilidad de conjuntos de datos en la aplicación. Por ejemplo, supongamos que la entidad cliente es una empresa con dos delegaciones, una en Madrid y otra en Barcelona. Se desea que la creación y gestión de usuario esté centralizada en Barcelona y que los usuarios de cada sede vean sólo la agenda y el directorio de su localidad. En este caso se crearía un dominio para la empresa y un área de trabajo para cada delegación.

Grupo Cada usuario pertenece a uno a más grupos. Es posible definir un número arbitrario de grupos. Los grupos en si mismos no otorgan roles a los usuarios. Es preciso asignar un rol al grupo dentro de una aplicación y área de trabajo para que los usuarios adquieran el rol deseado.

Gráfico ilustrativo del modelo de seguridad



© KnowGate 2010 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>

Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.