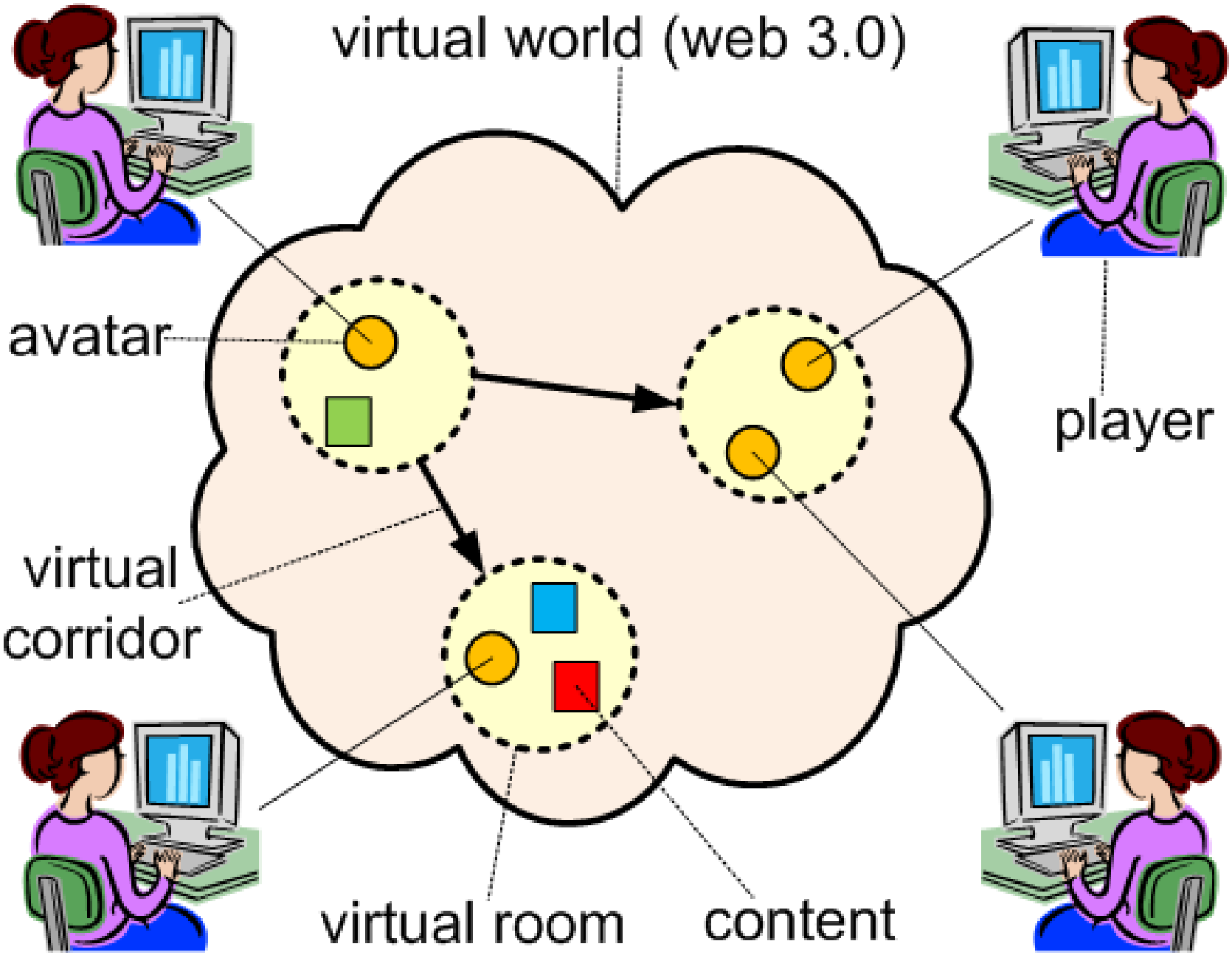
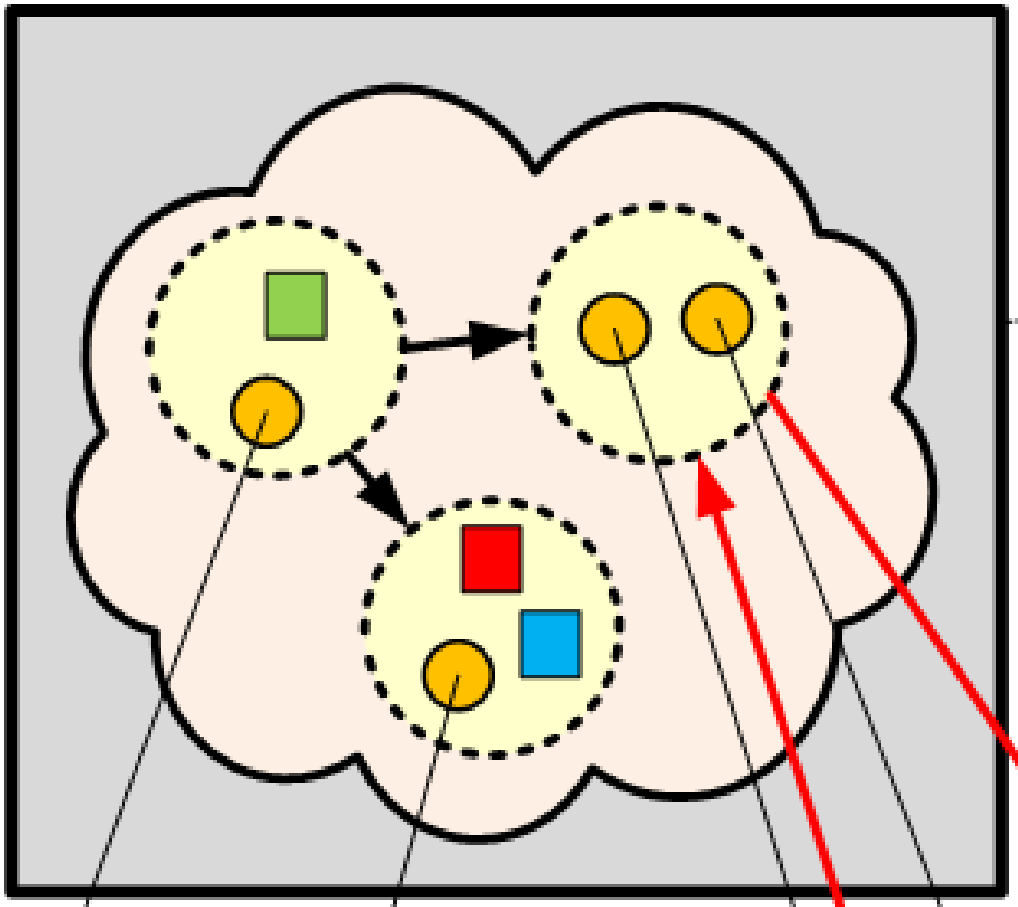
Using QuickSilver to mock-up Web 3.0

Krzys Ostrowski

Introduction

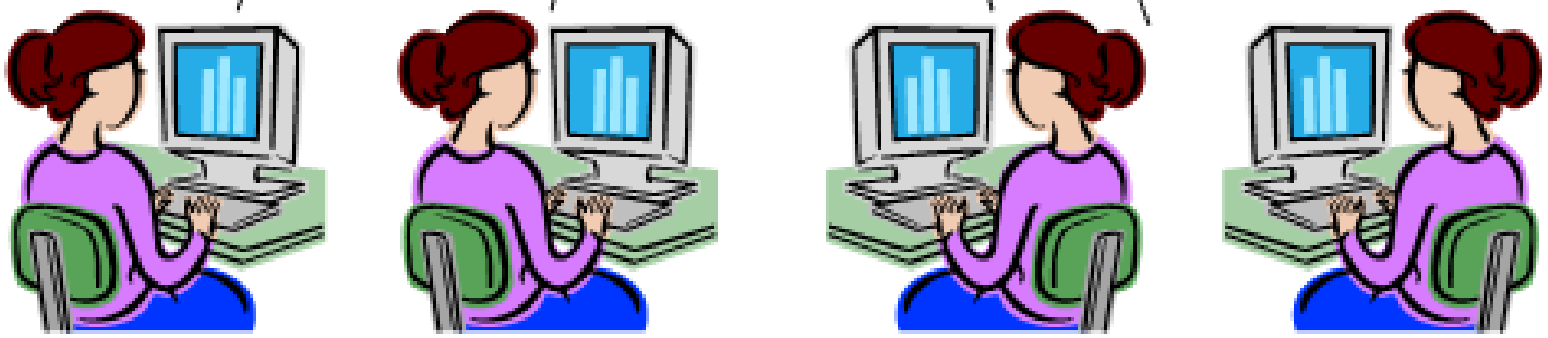

virtual world (web 3.0)
avatar
player
virtual
corridor
virtual room
content

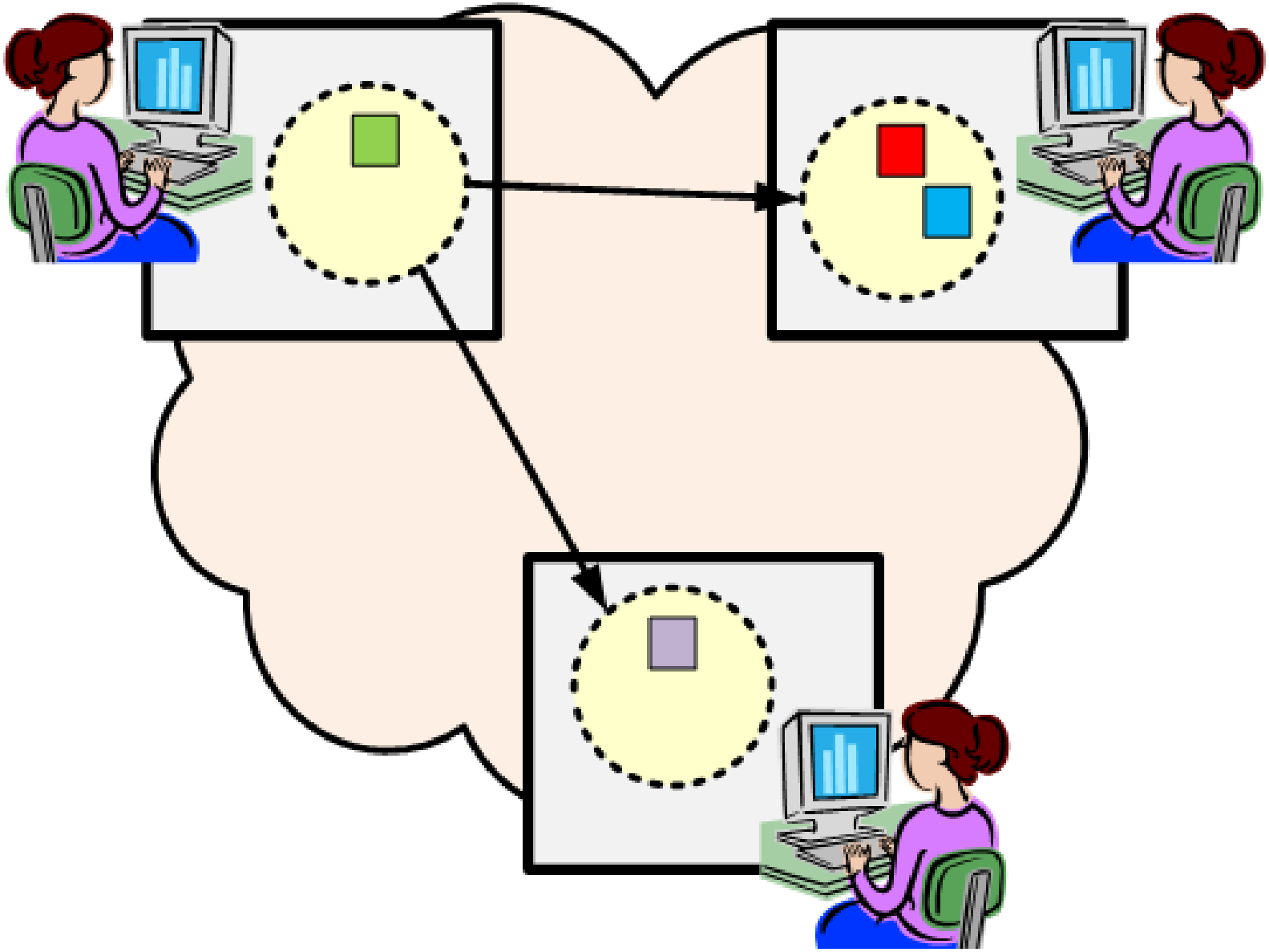
BAD



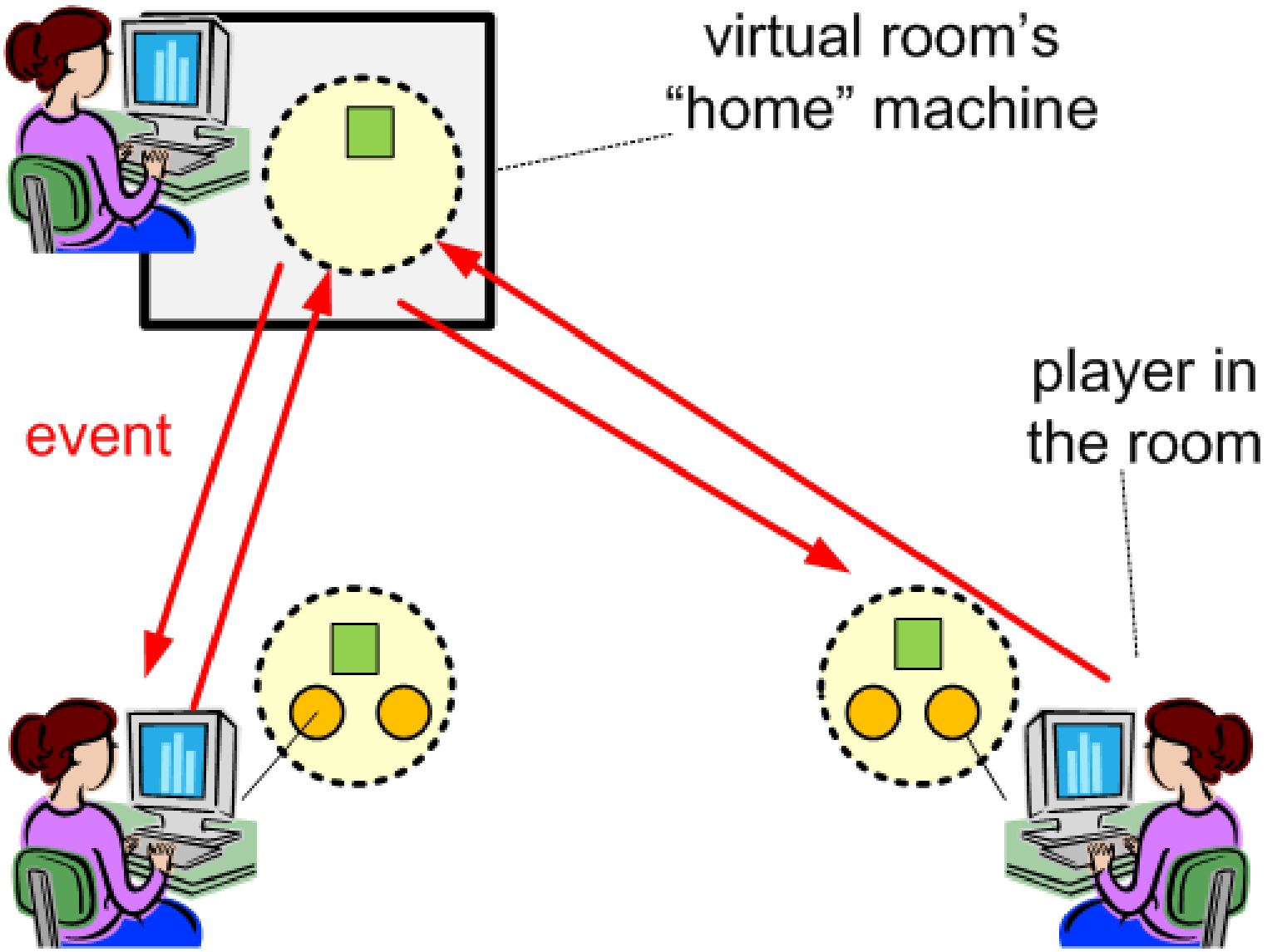
$100m
big,
scary
and
ugly
server
farm

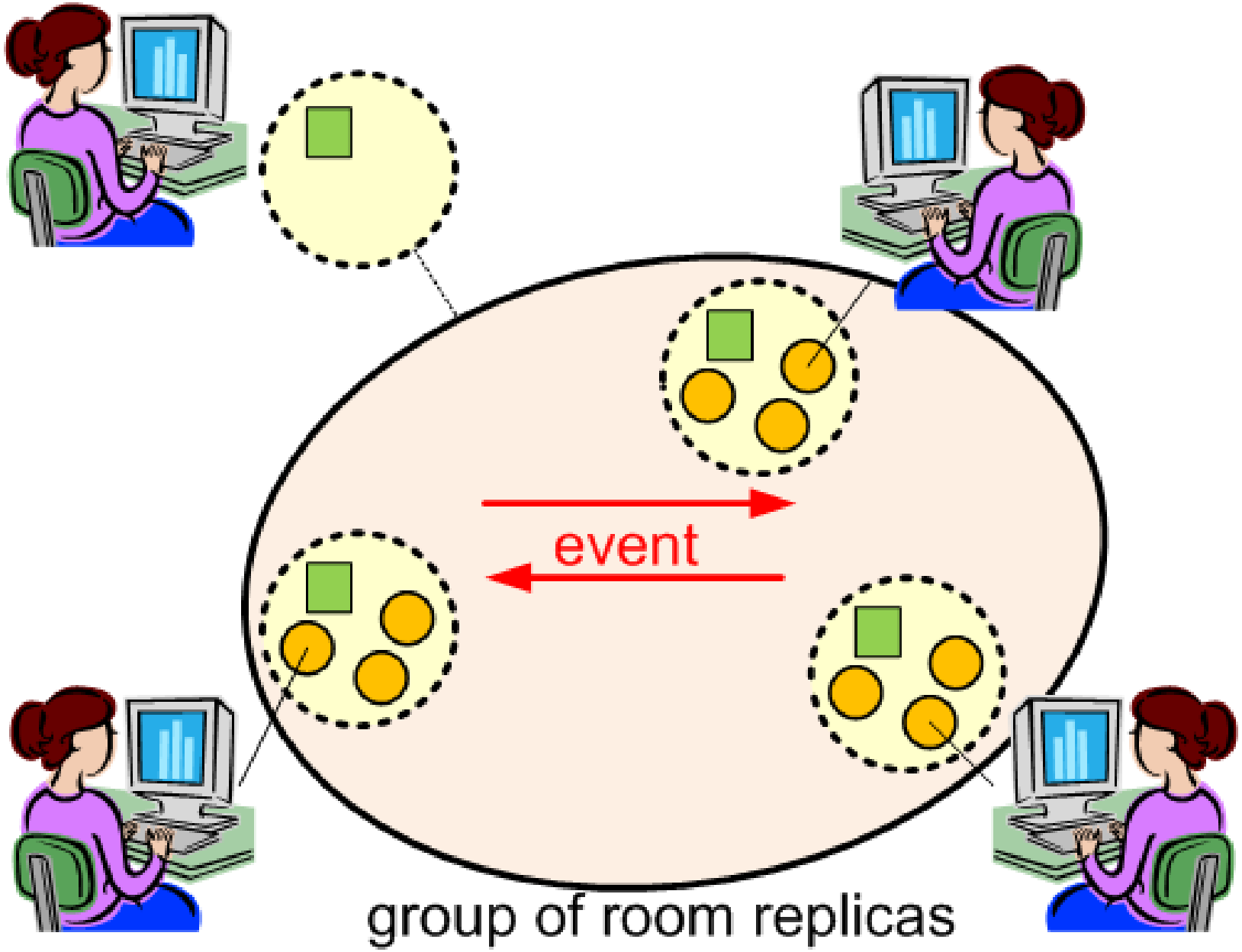
event

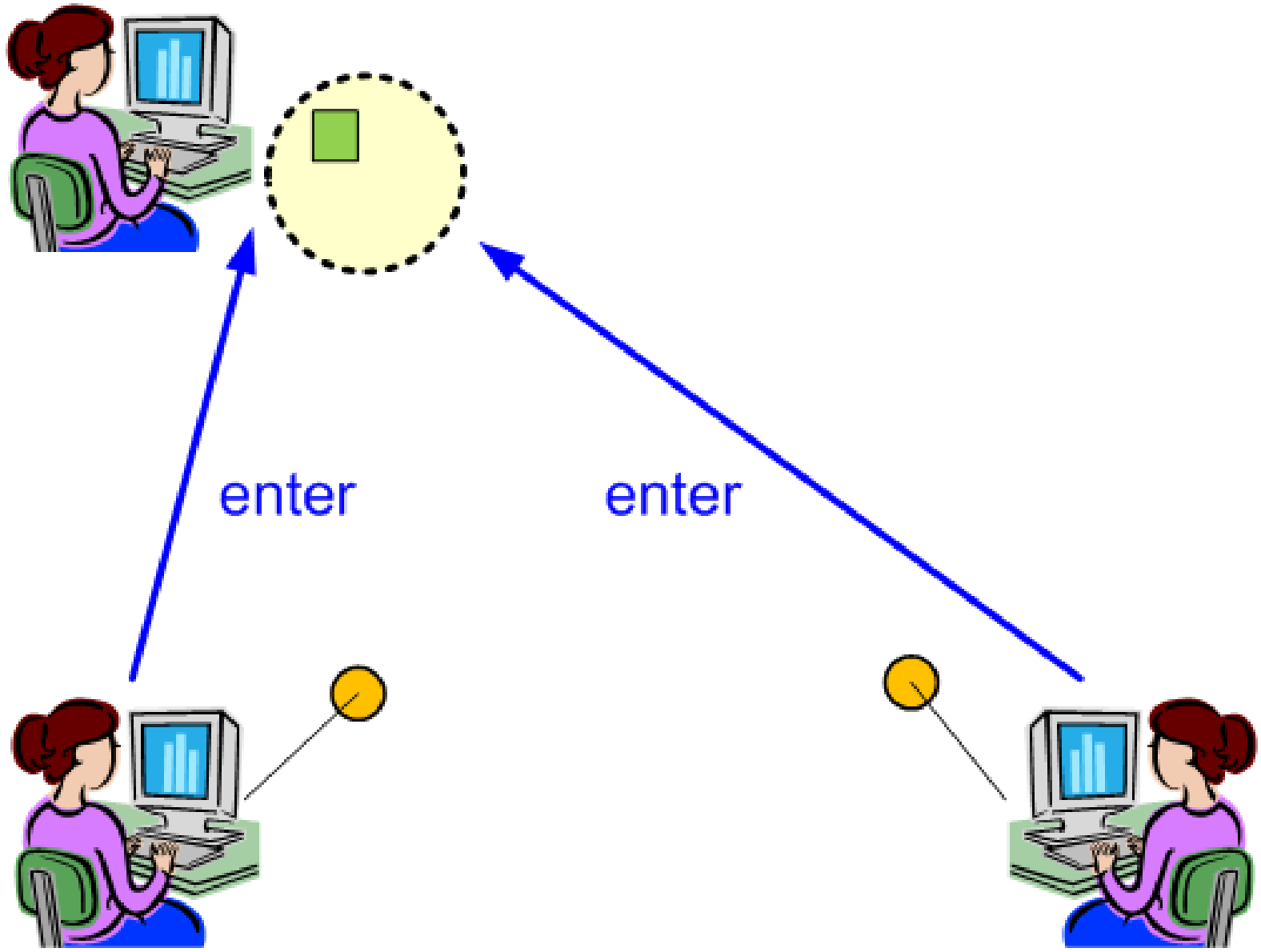


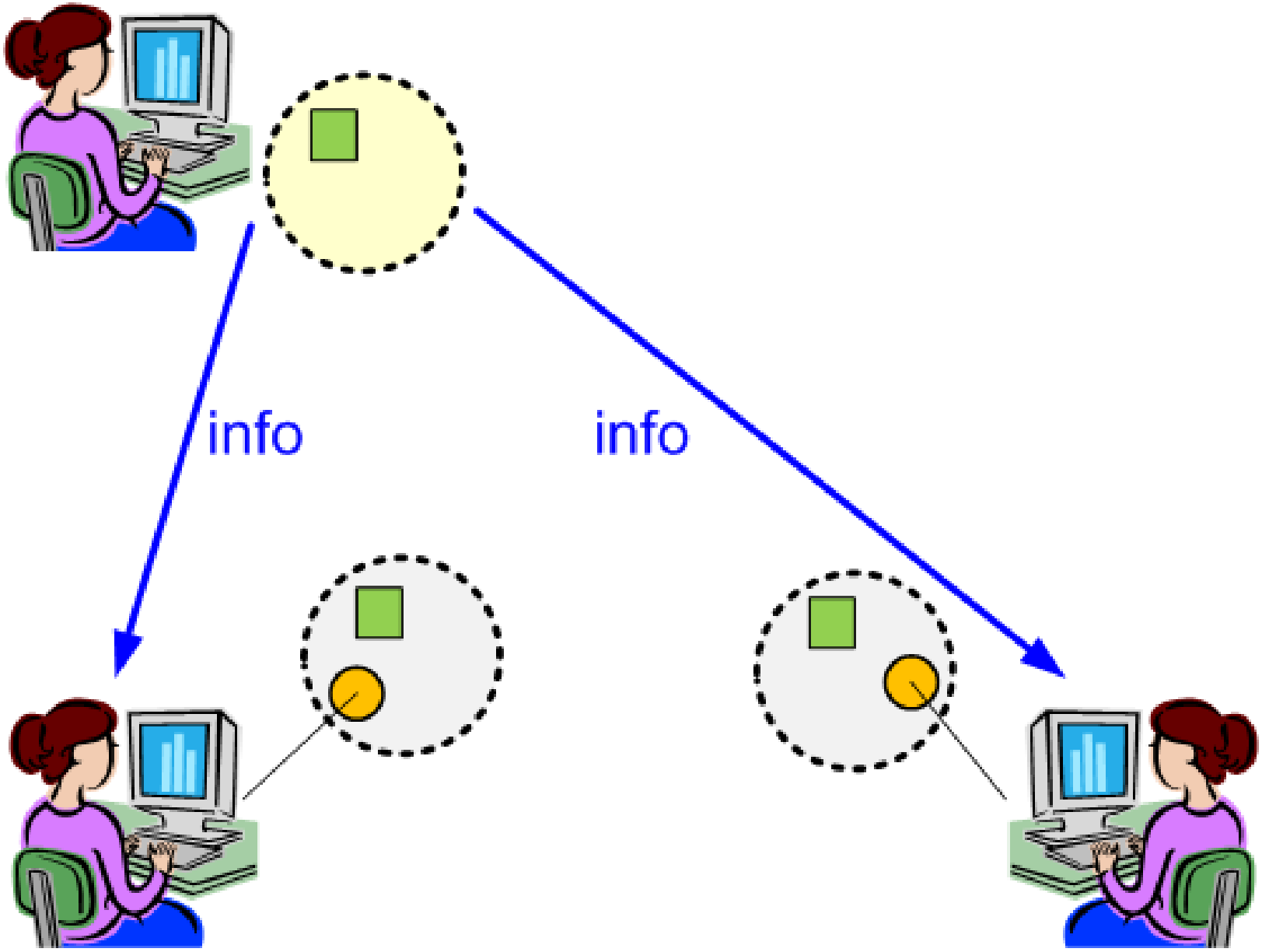


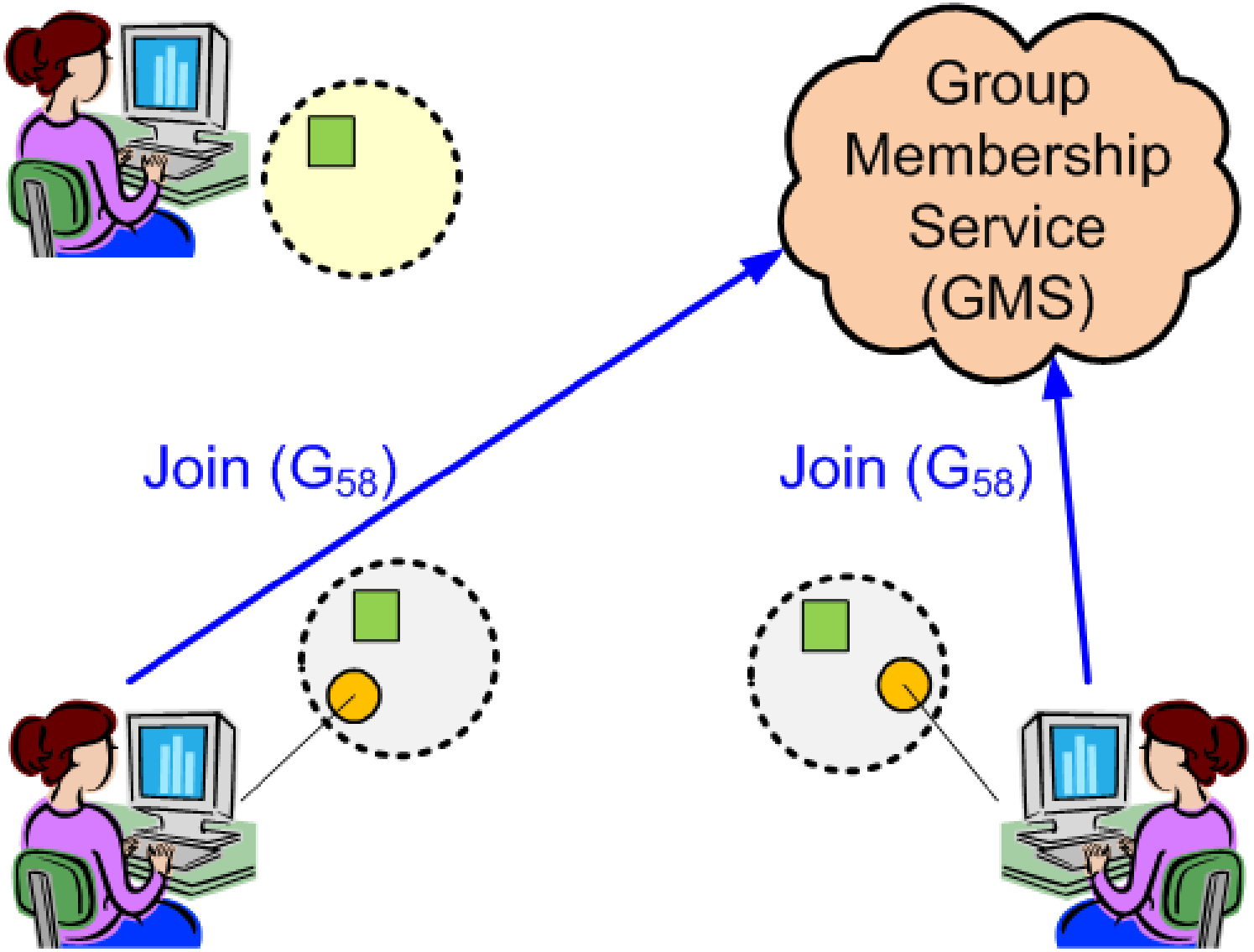
BAD

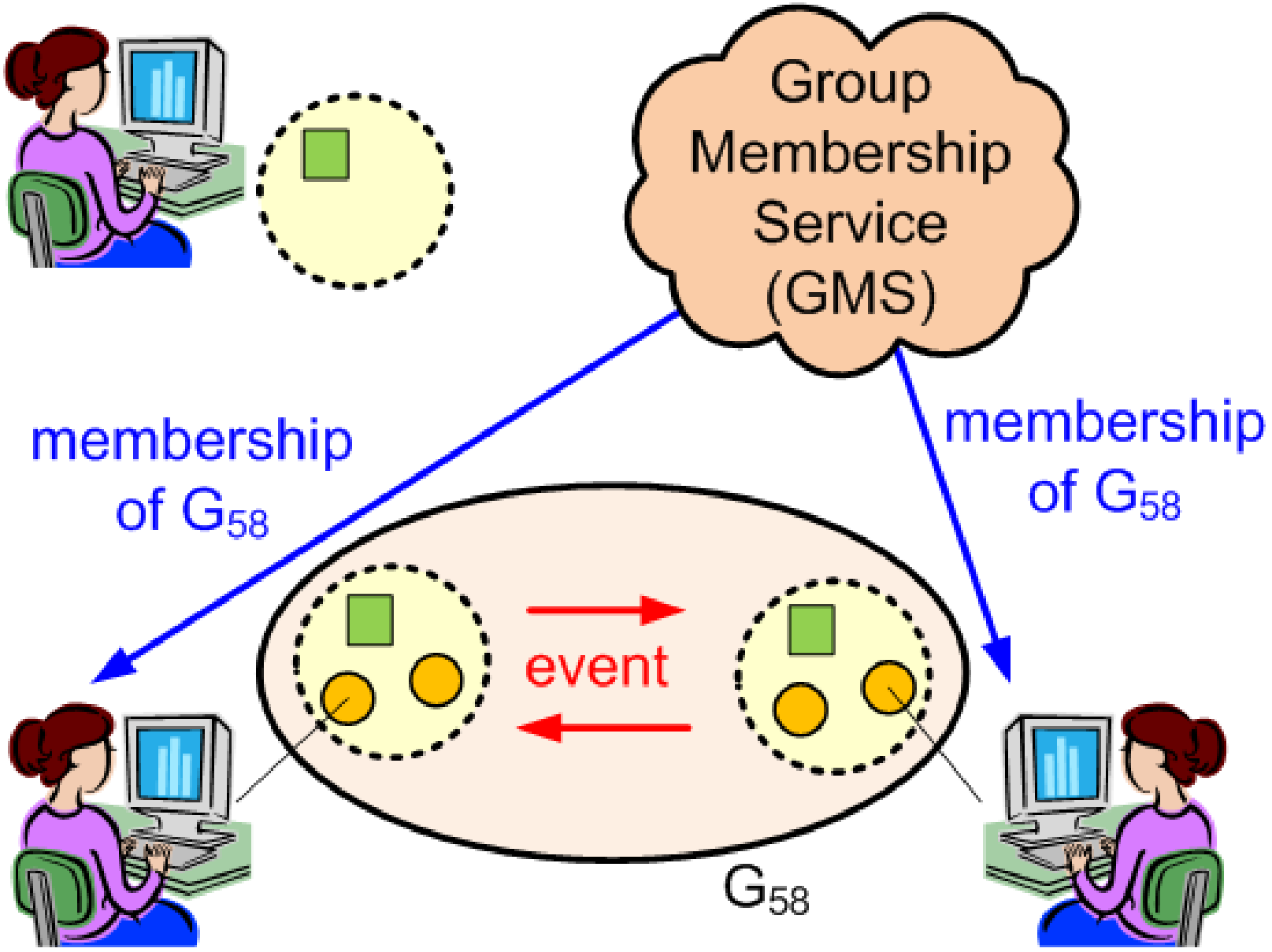


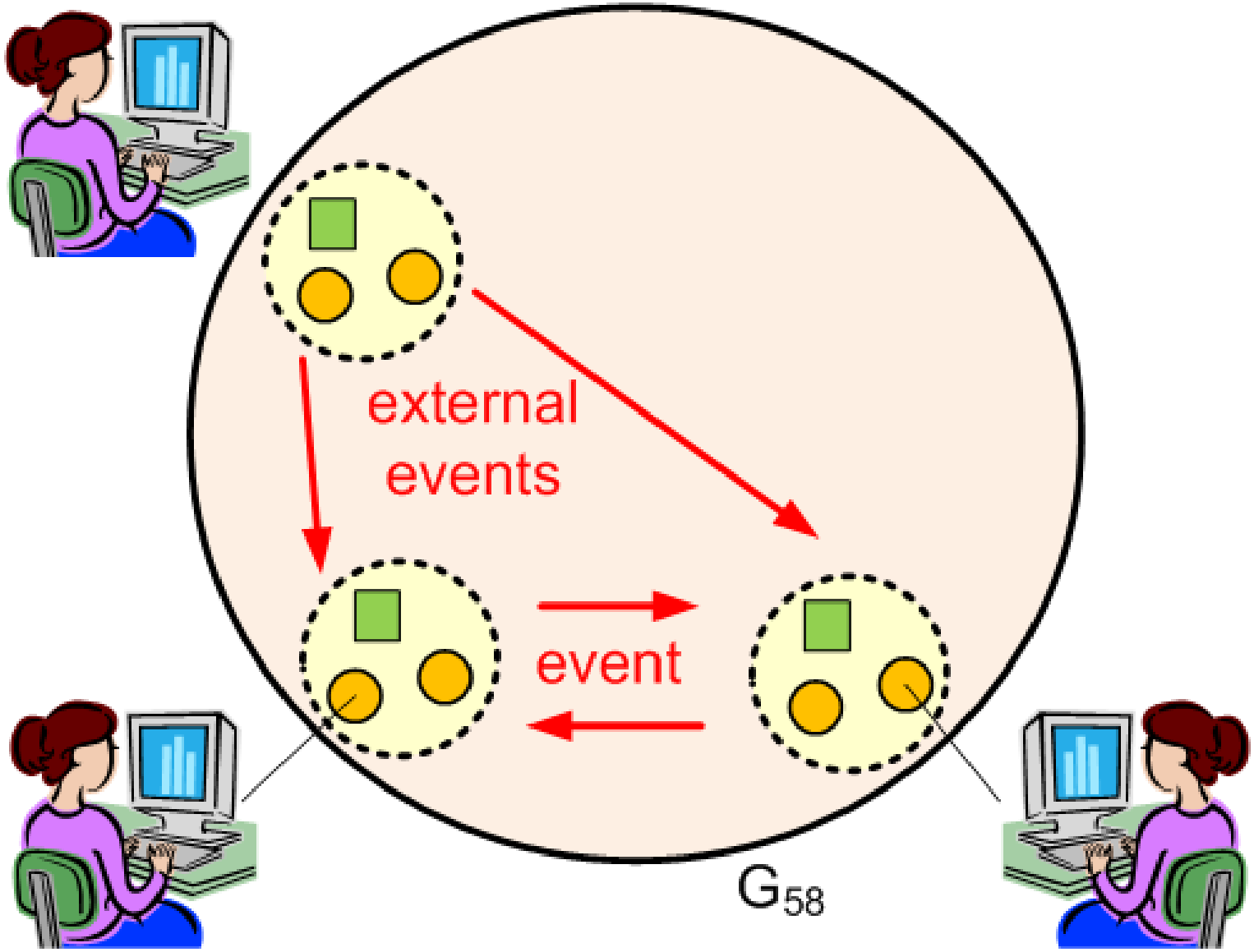

event
group of room replicas


enter
enter

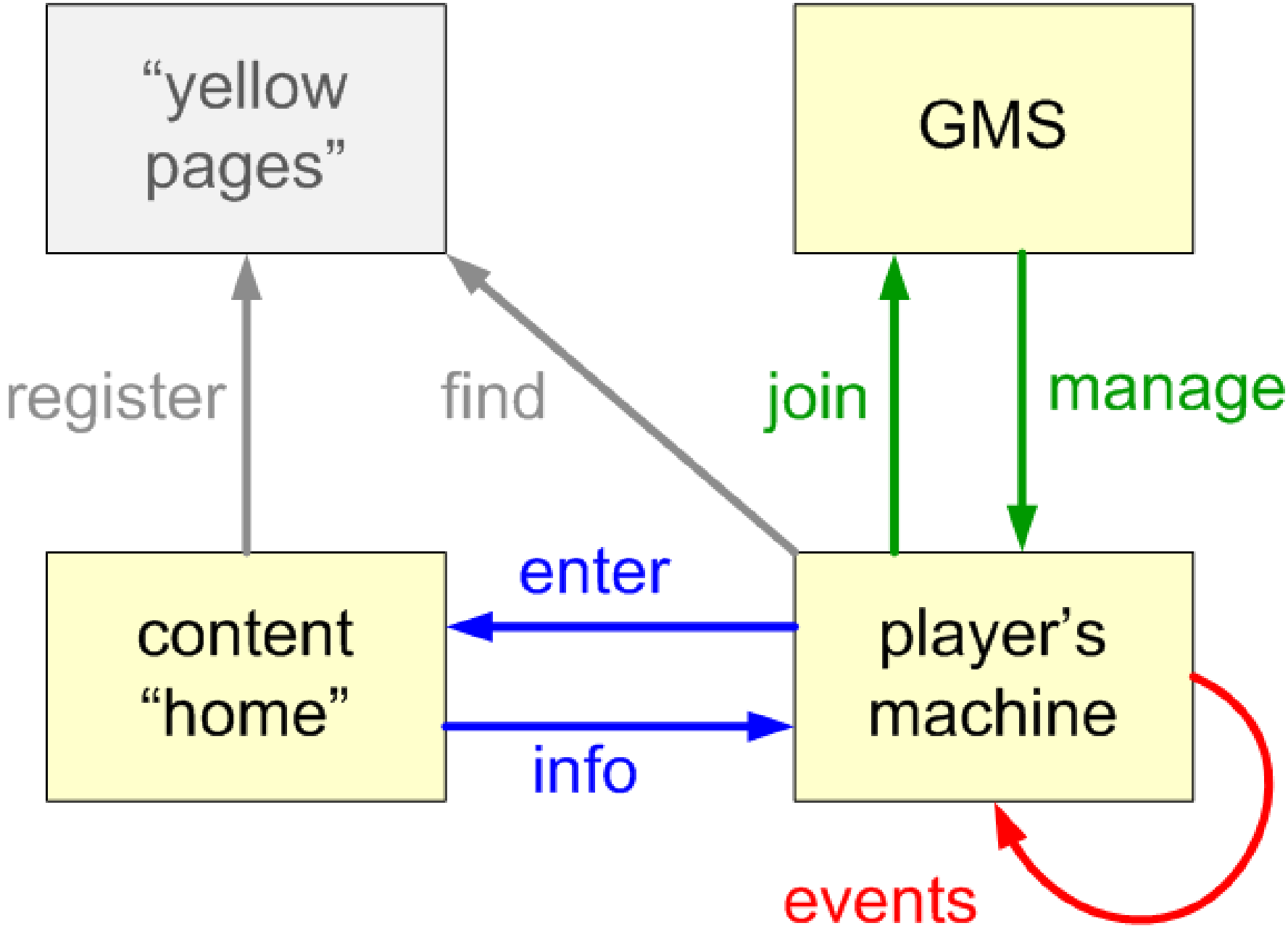

info
info

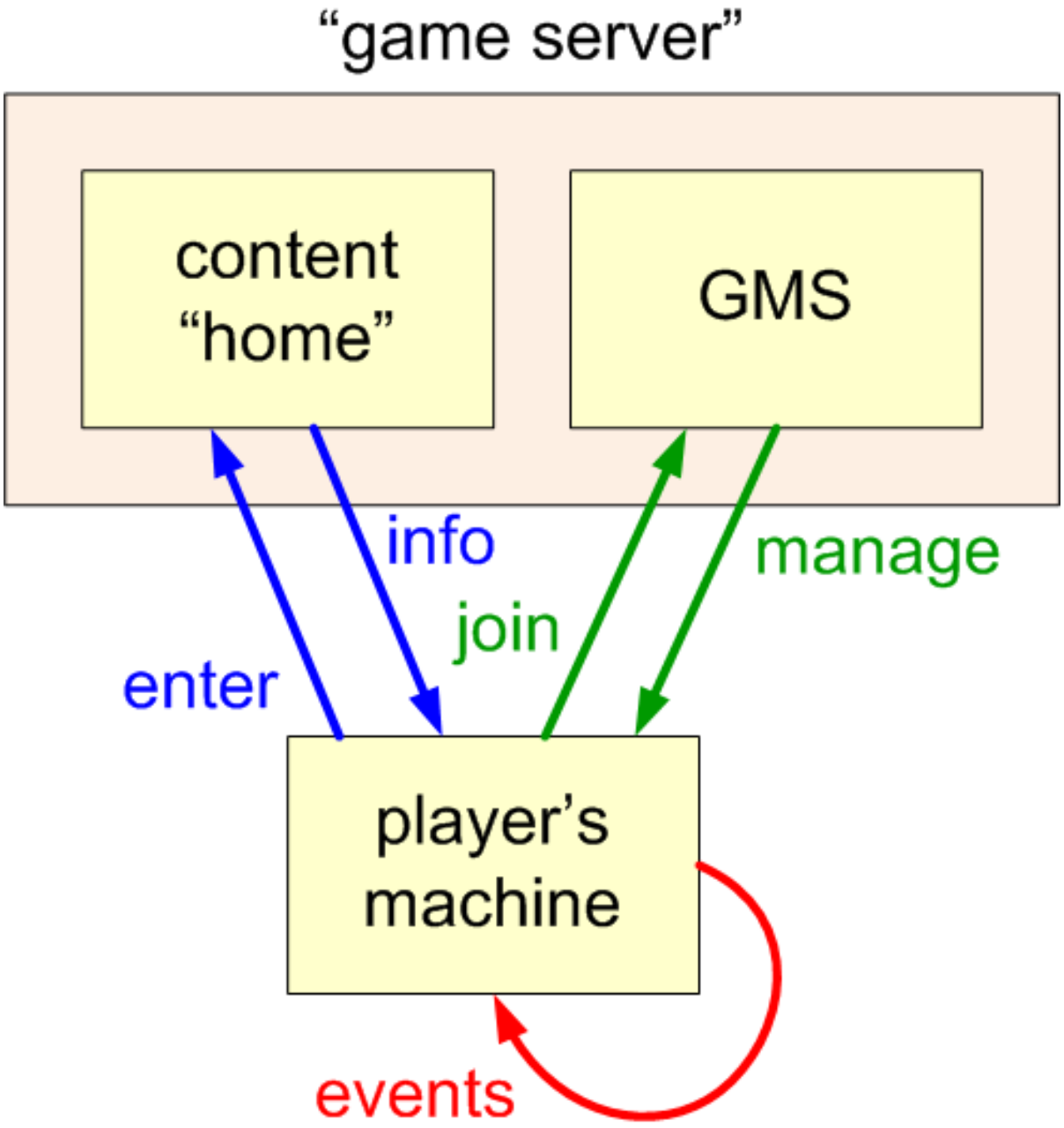

Group
Membership
Service
(GMS)
Join (G_{58})
Join (G_{58})

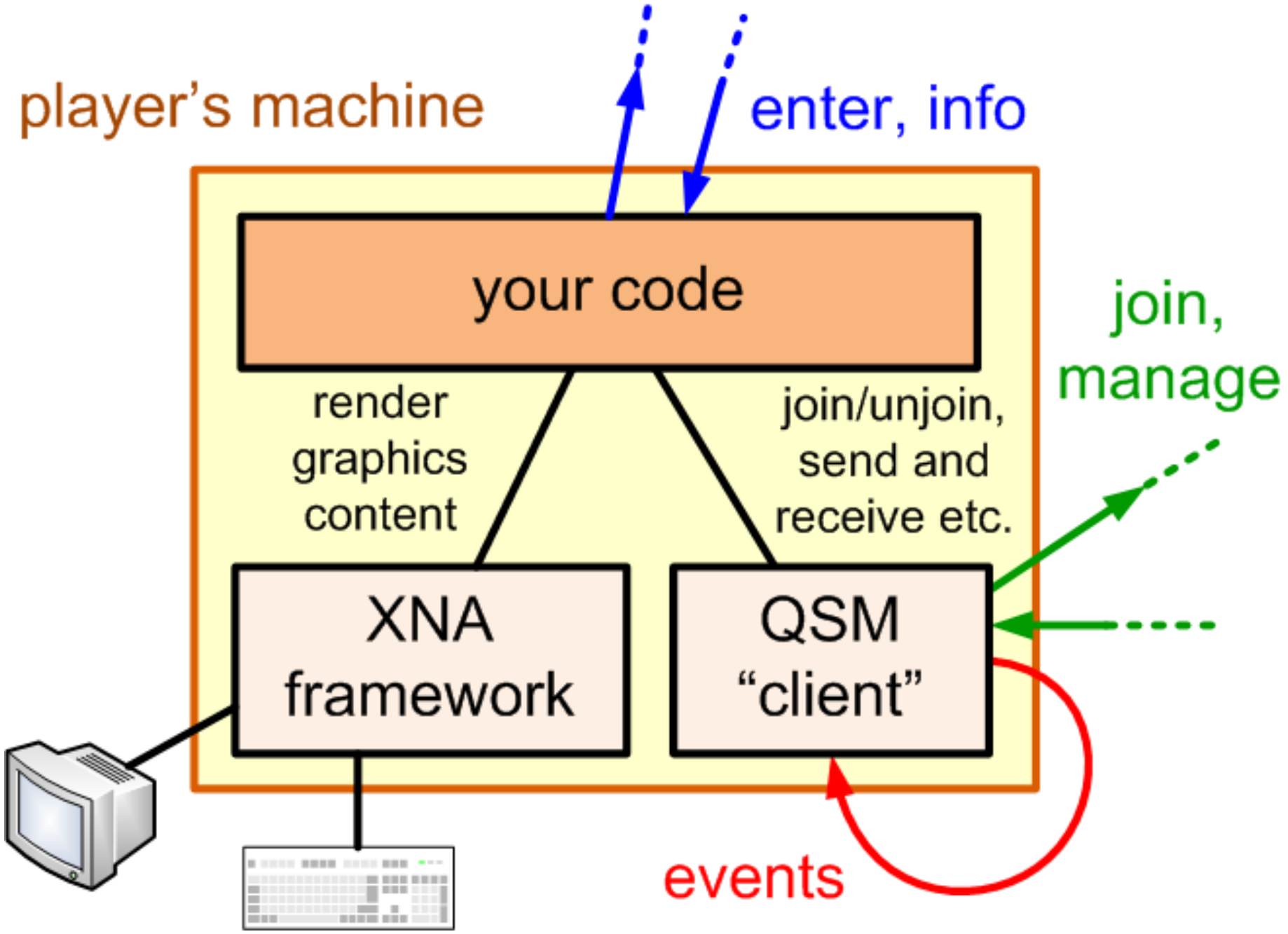

Group
Membership
Service
(GMS)
membership
of G_{58}
membership
of G_{58}
event
G_{58}

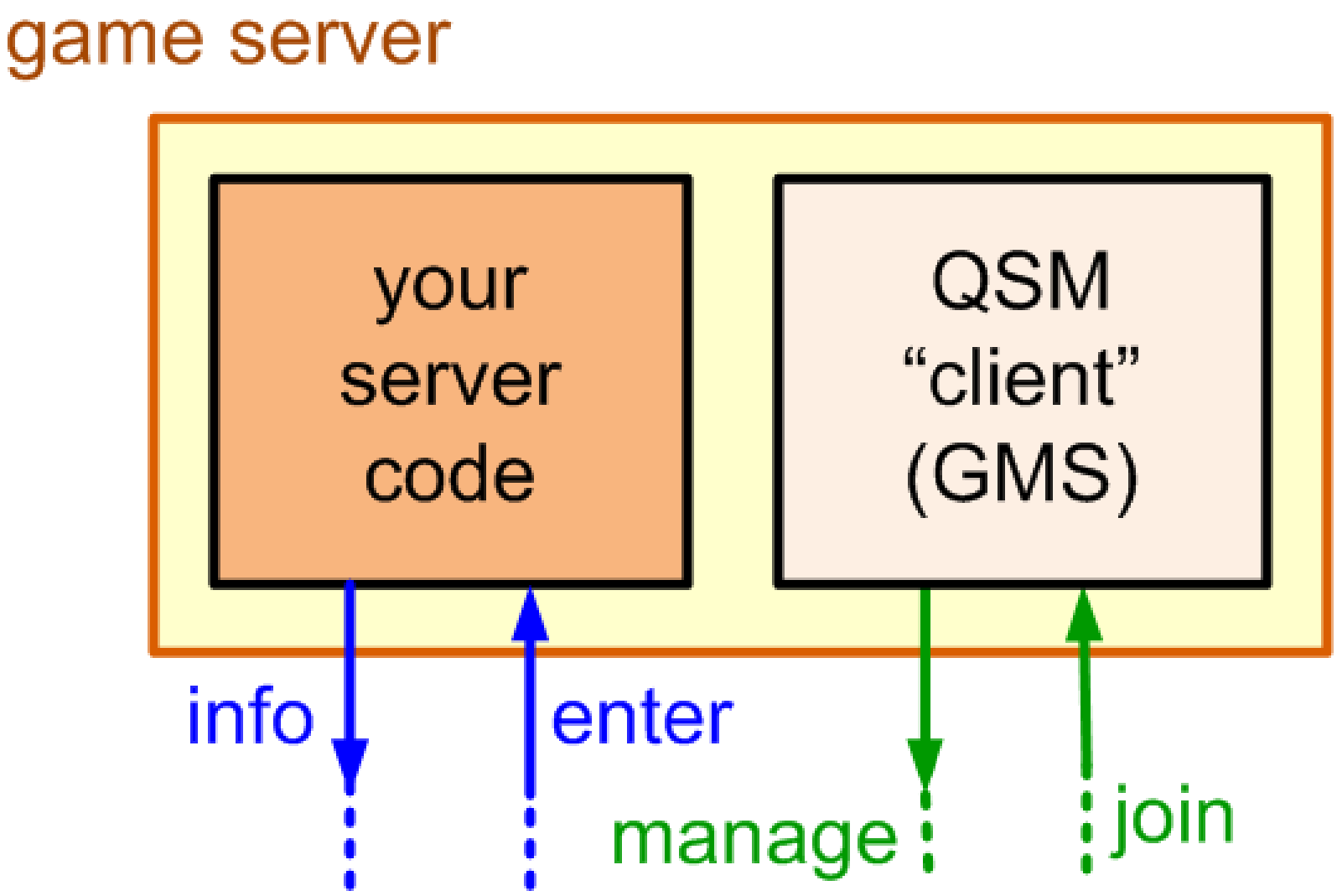

external
events
event

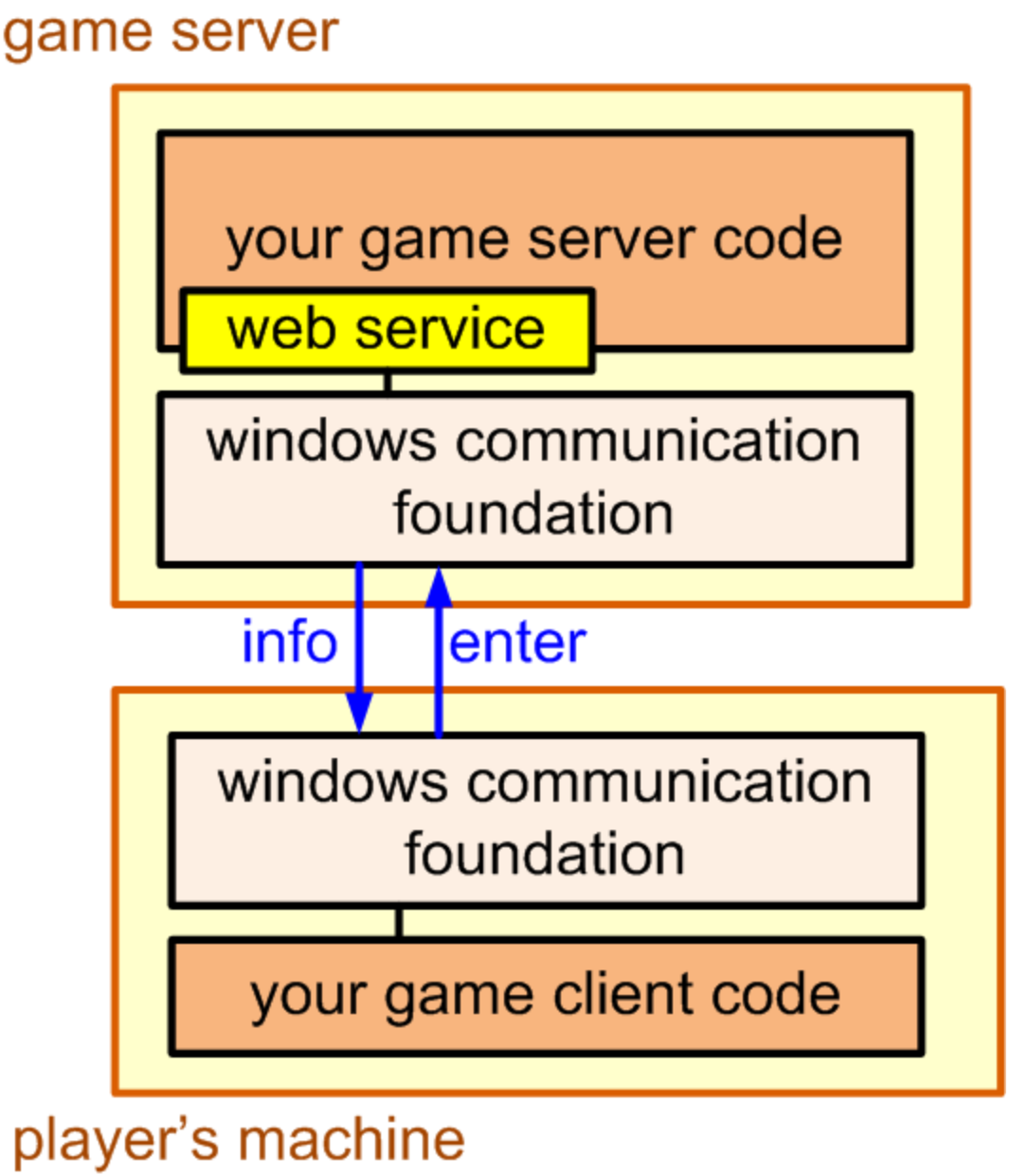
Architecture


"yellow pages"
GMS
register
find
join
manage
enter
content "home"
player's machine
info
events


“game server”
content
“home”
GMS
info
join
manage
enter
player’s
machine
events


player's machine
enter, info
your code
join,
manage
render
graphics
content
join/unjoin,
send and
receive etc.
XNA
framework
QSM
"client"
events


game server
your
server
code
QSM
“client”
(GMS)
info
enter
manage
join


game server
your game server code
web service
windows communication foundation
info
enter
windows communication foundation
your game client code
player's machine

Tools

Development Environment

- **Microsoft Visual Studio 2005 “Express”**

  (supports XNA Framework & Game Studio Express)

  – XNA Project types

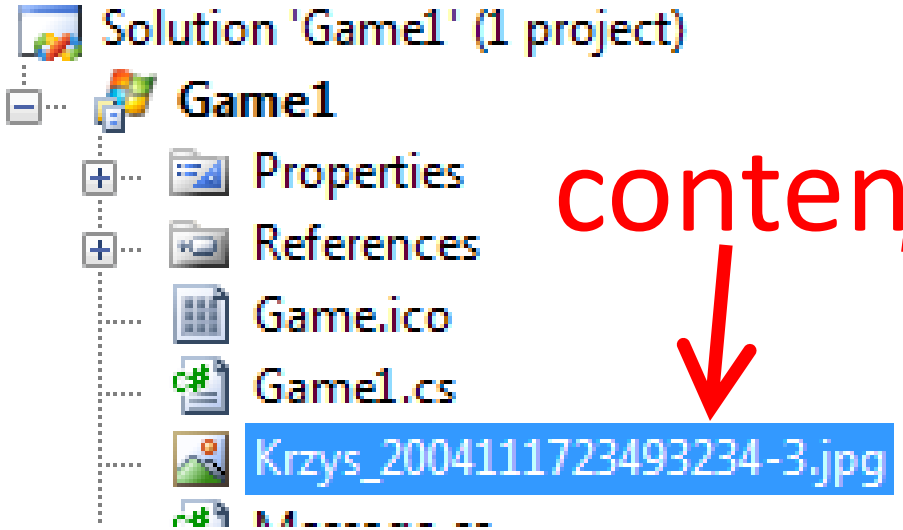
  – Implements the “content pipeline”

- **Microsoft Visual Studio 2005 “Professional”**

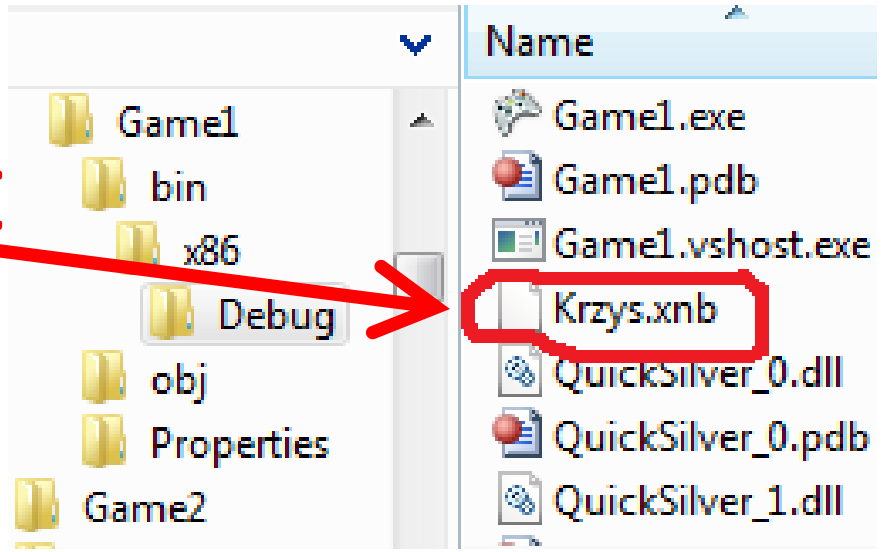
  (support only the XNA Framework)

  – Version control,

  – Debugging multithreaded applications, etc.



content

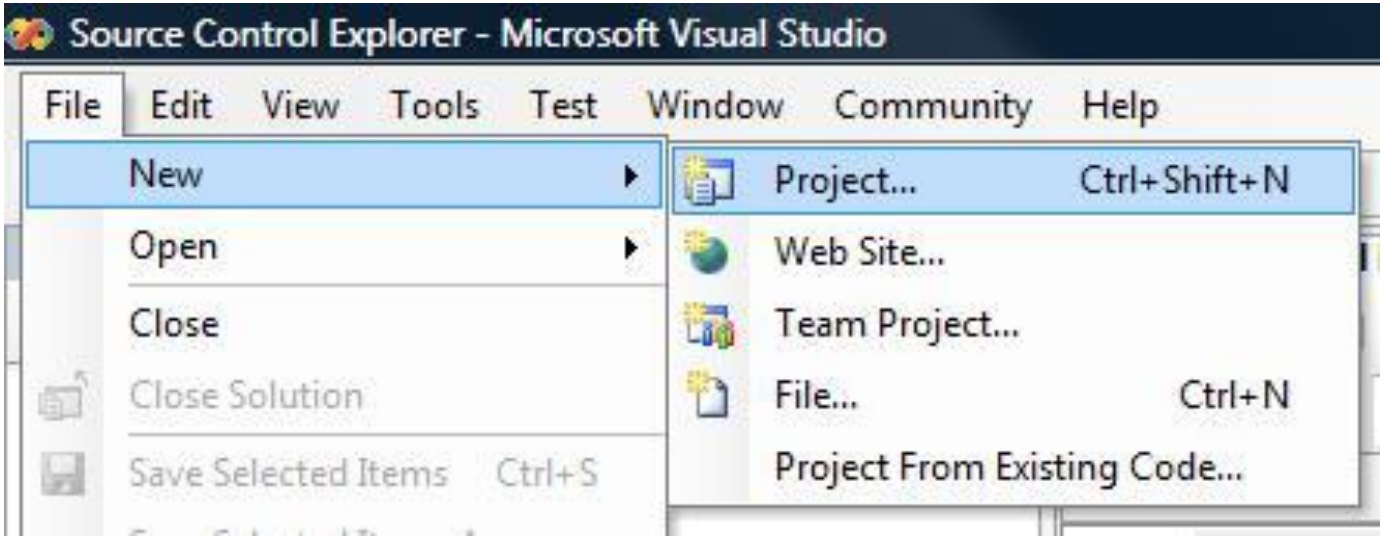


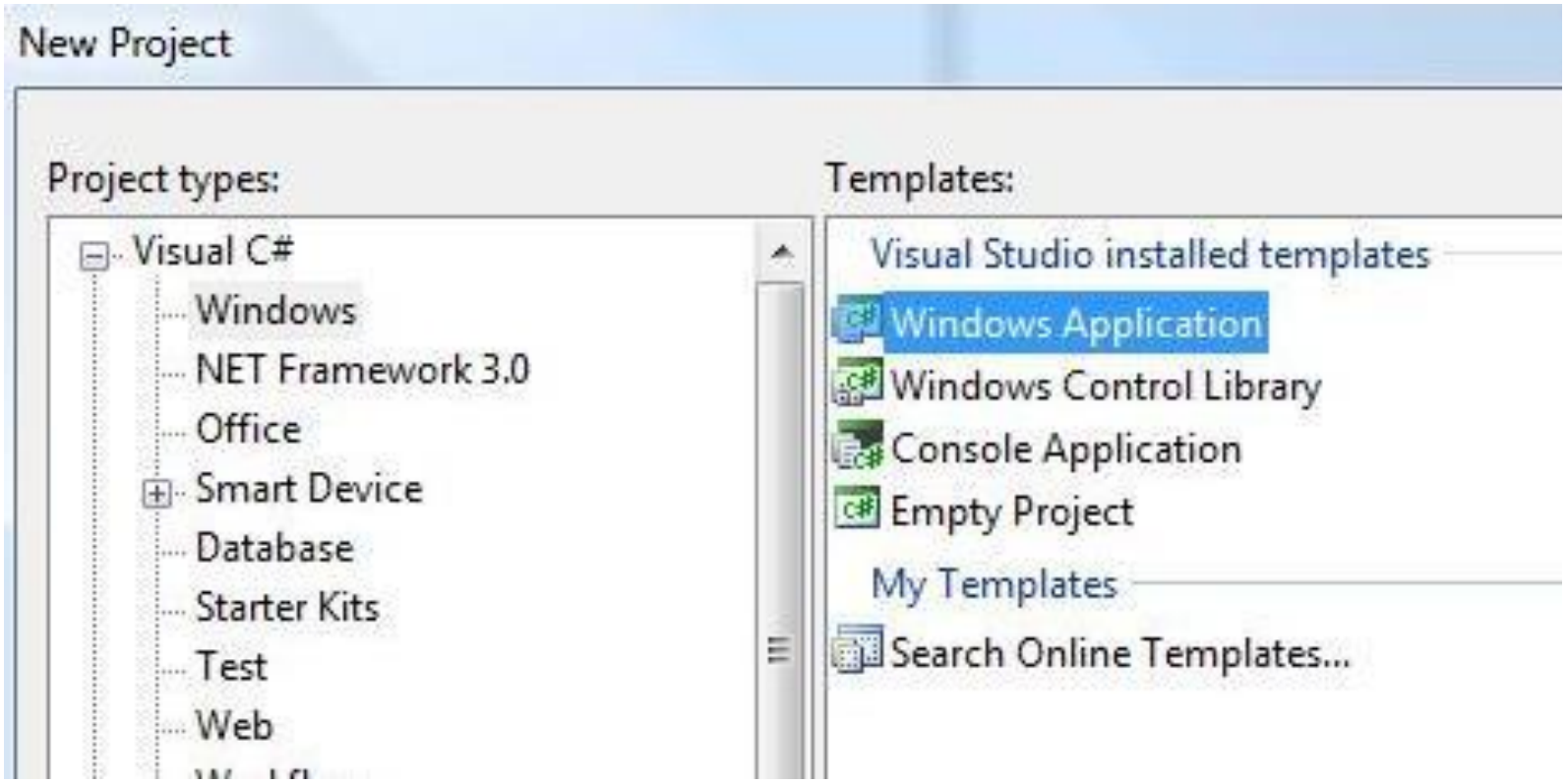
loading content

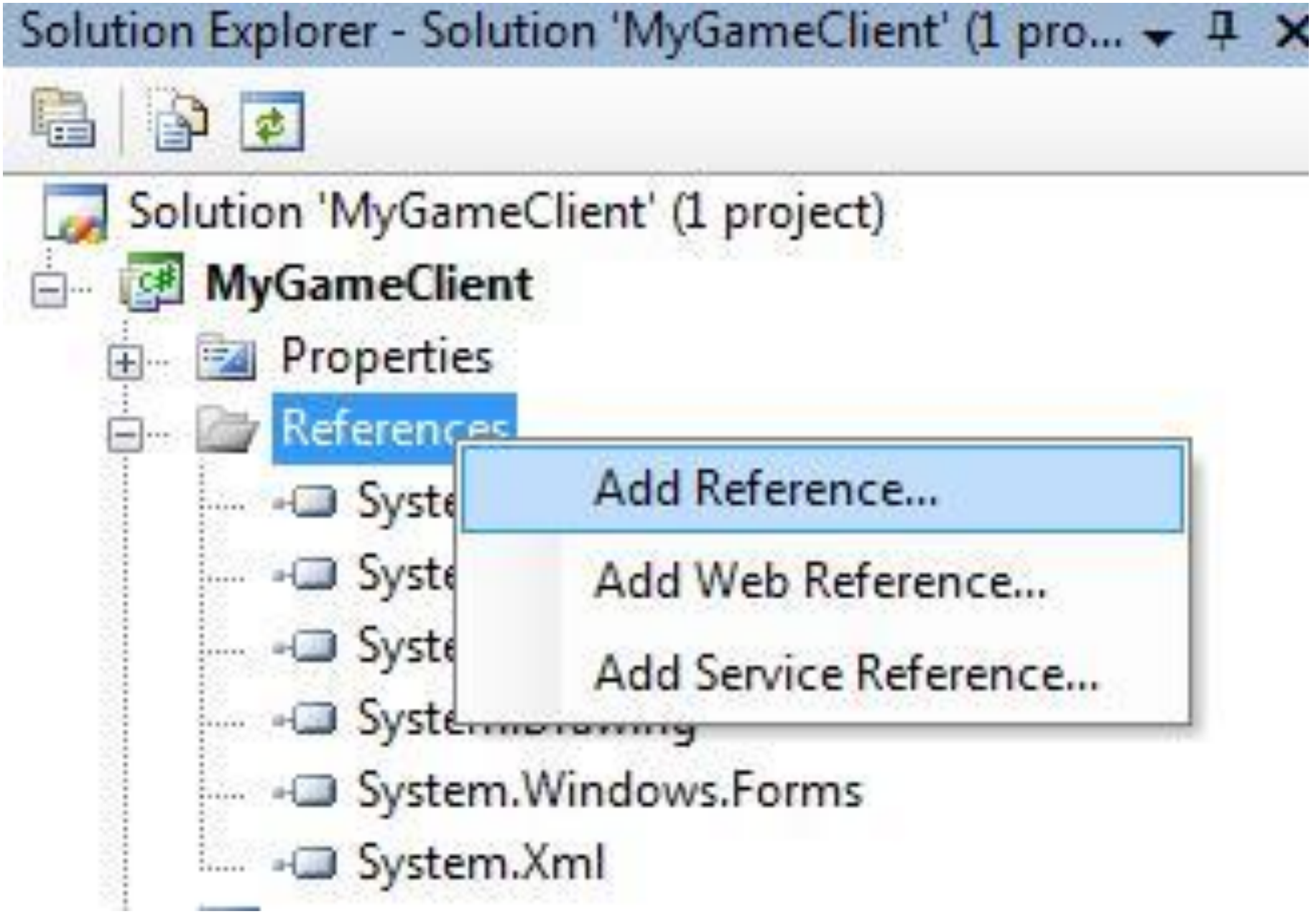
```
protected override void LoadGraphicsContent(bool loadAllContent)
{
    if (loadAllContent)
    {
        // TODO: Load any ResourceManagementMode.Automatic content
        myTexture = content.Load<Texture2D>("Krzys");
        player = CreatePlayer(n, Vector2.Zero);
    }
```

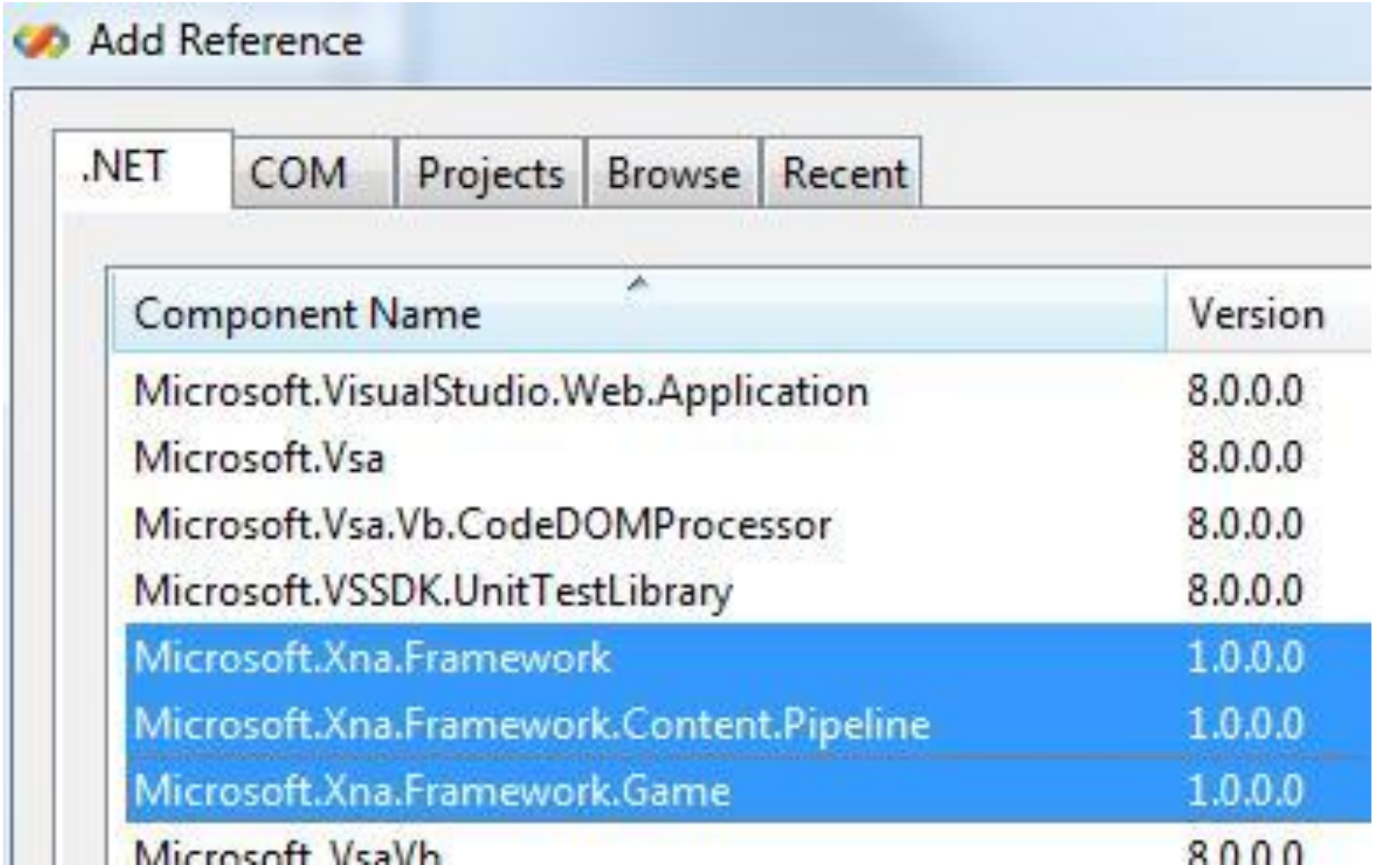
Coding

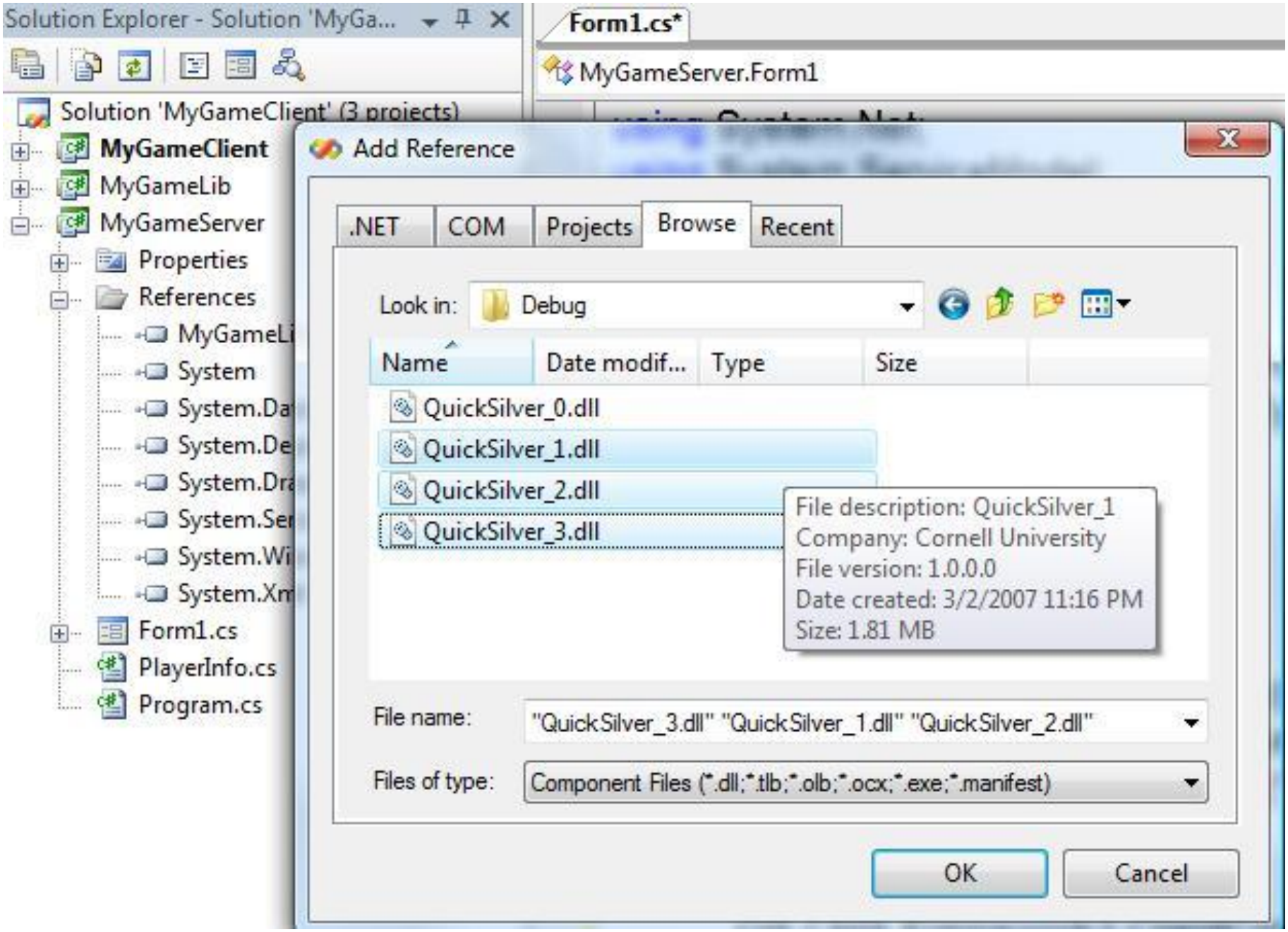
How to access the functionality provided by XNA, QMS etc.

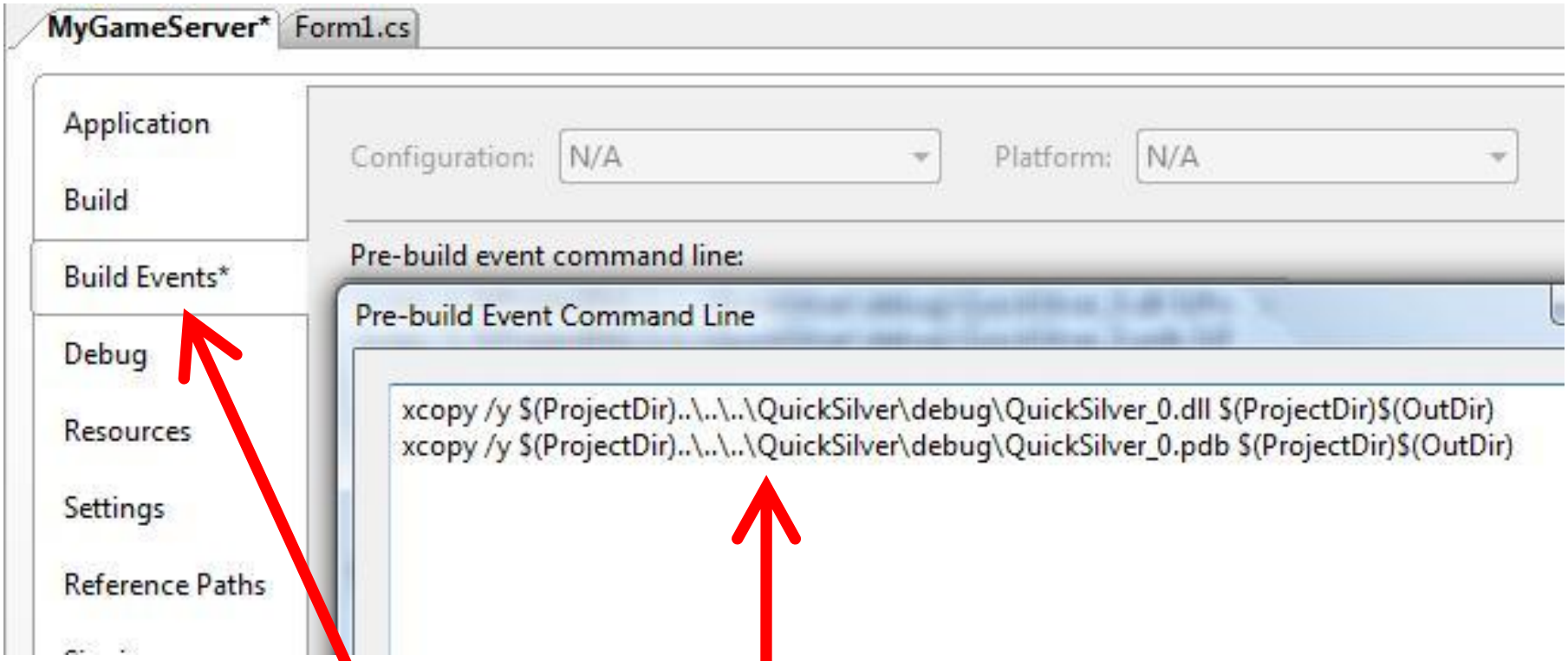

Source Control Explorer - Microsoft Visual Studio
File
Edit
View
Tools
Test
Window
Community
Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Project... Ctrl+Shift+N
Web Site...
Team Project...
File... Ctrl+N
Project From Existing Code...


New Project
Project types:
Visual C#
Windows
NET Framework 3.0
Office
Smart Device
Database
Starter Kits
Test
Web
Templates:
Visual Studio installed templates
Windows Application
Windows Control Library
Console Application
Empty Project
My Templates
Search Online Templates...


Solution Explorer - Solution 'MyGameClient' (1 pro...
Solution 'MyGameClient' (1 project)
MyGameClient
Properties
References
Add Reference...
Add Web Reference...
Add Service Reference...
System.Windows.Forms
System.Xml


Add Reference
.NET
COM
Projects
Browse
Recent
Component Name
Version
Microsoft.VisualStudio.Web.Application
8.0.0.0
Microsoft.Vsa
8.0.0.0
Microsoft.Vsa.Vb.CodeDOMProcessor
8.0.0.0
Microsoft.VSSDK.UnitTestLibrary
8.0.0.0
Microsoft.Xna.Framework
1.0.0.0
Microsoft.Xna.Framework.Content.Pipeline
1.0.0.0
Microsoft.Xna.Framework.Game
1.0.0.0

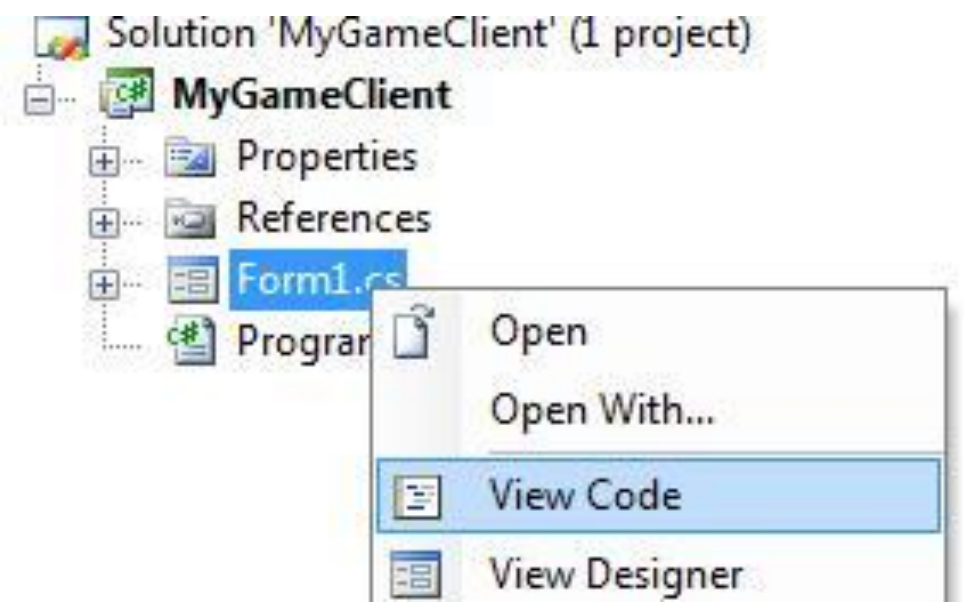

Solution Explorer - Solution 'MyGa...
Solution 'MyGameClient' (3 projects)
MyGameClient
MyGameLib
MyGameServer
Properties
References
MyGameLi
System
System.Da
System.De
System.Dra
System.Ser
System.Wi
System.Xm
Form1.cs
PlayerInfo.cs
Program.cs
Form1.cs*
MyGameServer.Form1
Add Reference
.NET
COM
Projects
Browse
Recent
Look in:
Debug
Name
Date modif...
Type
Size
QuickSilver_0.dll
QuickSilver_1.dll
QuickSilver_2.dll
QuickSilver_3.dll
File description: QuickSilver_1
Company: Cornell University
File version: 1.0.0.0
Date created: 3/2/2007 11:16 PM
Size: 1.81 MB
File name:
"QuickSilver_3.dll" "QuickSilver_1.dll" "QuickSilver_2.dll"
Files of type:
Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
OK
Cancel



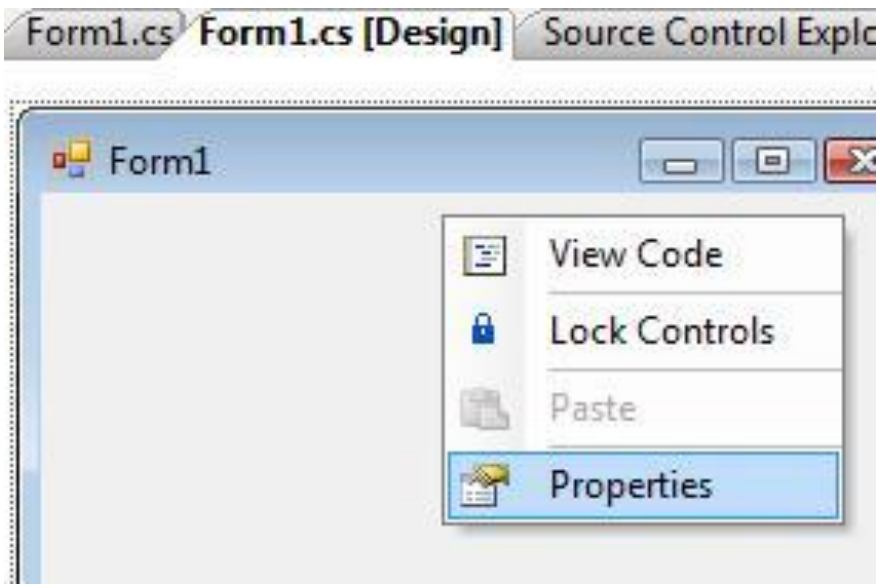
Add a pre-build event to manually copy QuickSilver_0.dll,

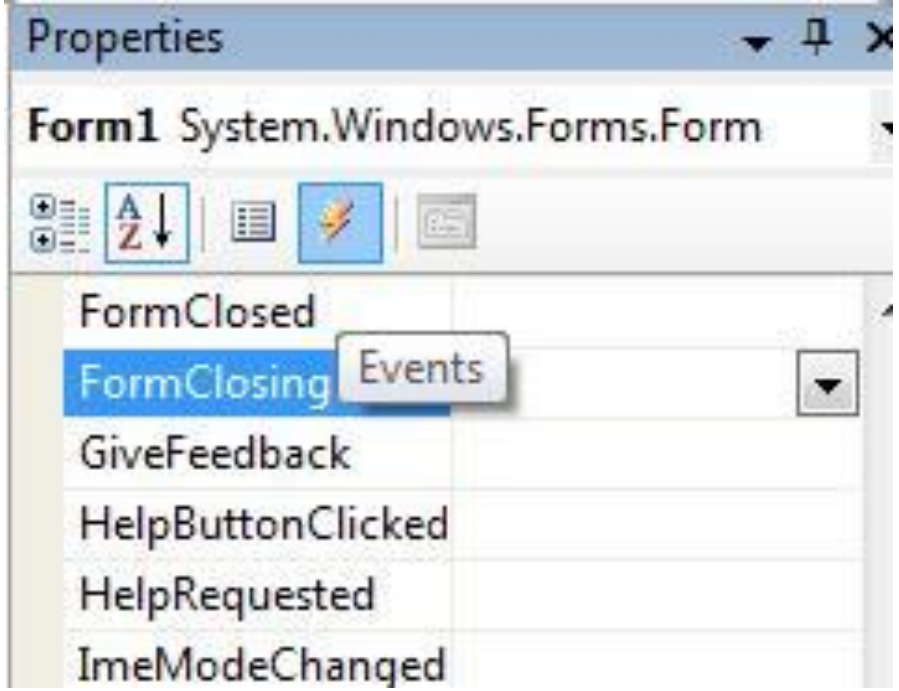
You can also copy over graphical content here

Where to write your own code

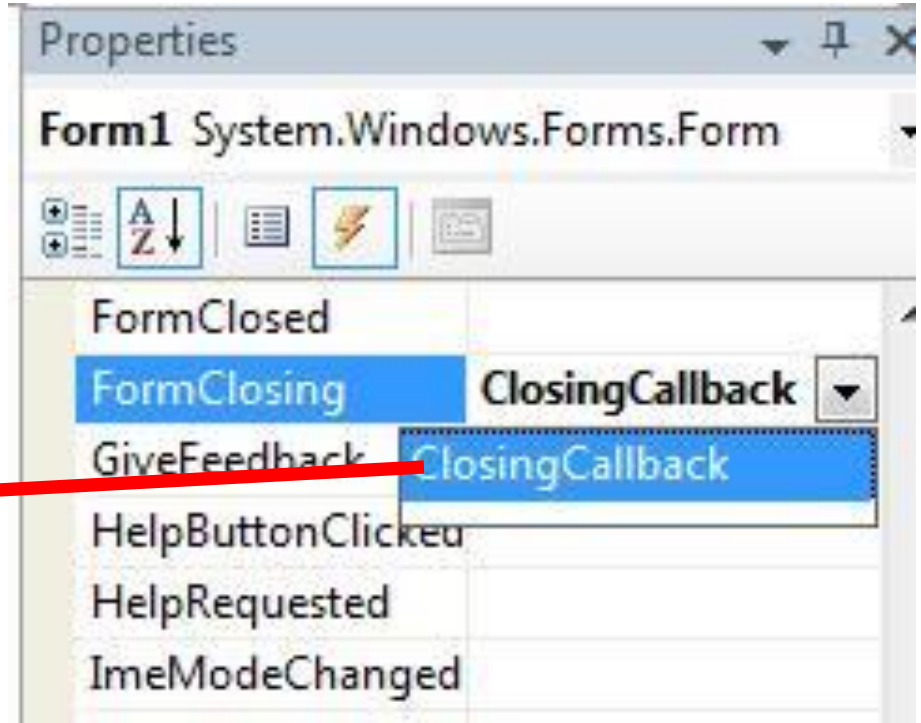


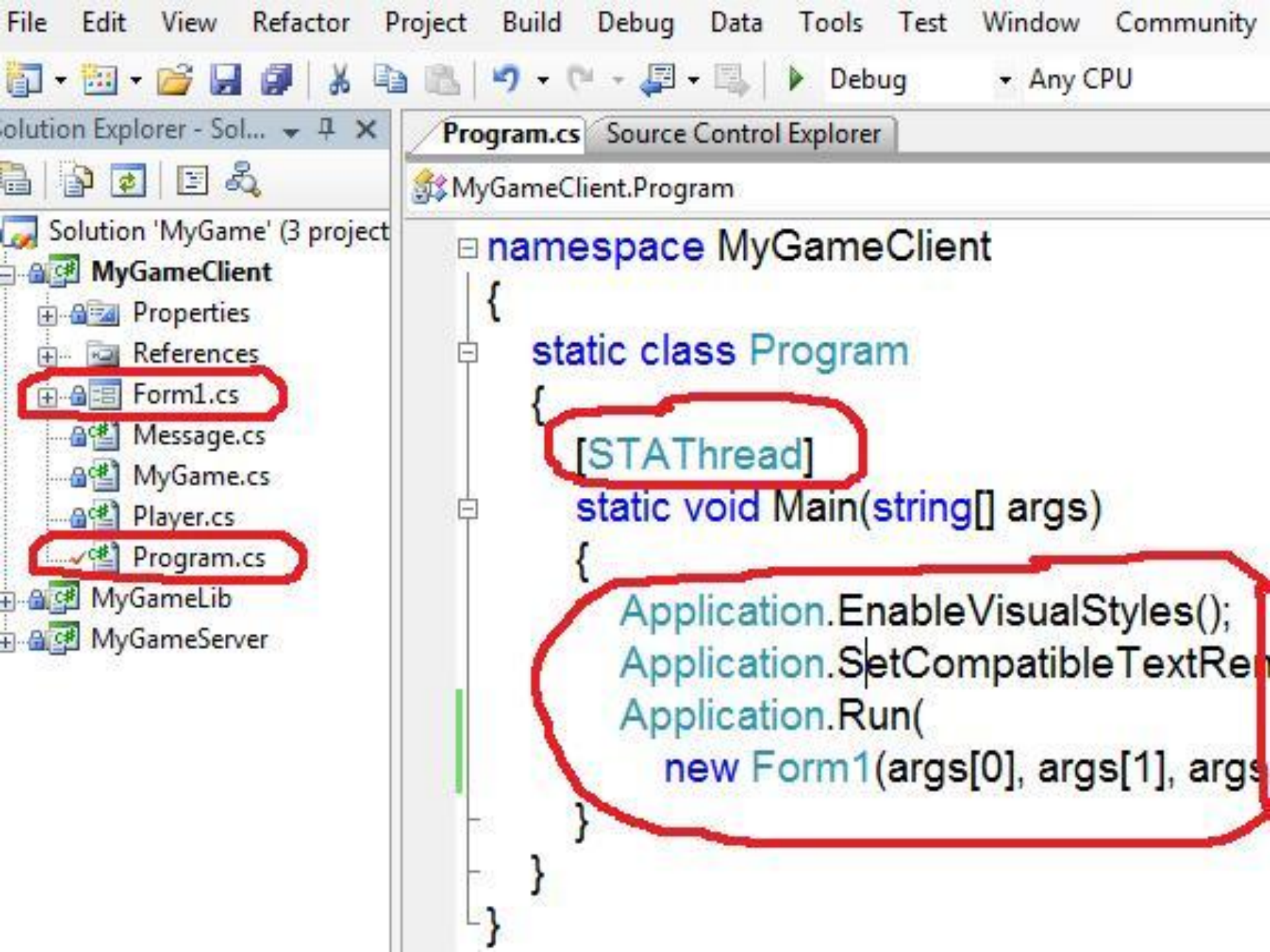

```
namespace MyGameClient
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
         gamethread = new Thread(new ThreadStart(ThreadCallback));
         gamethread.Start();
         …
```






```
public partial class Form1 : Form
{
   …

   private void ClosingCallback(
      object sender, FormClosingEventArgs e)
   {
      …
   }
```




File Edit View Refactor Project Build Debug Data Tools Test Window Community
Debug
Any CPU
Solution Explorer - Sol...
Solution 'MyGame' (3 project
MyGameClient
Properties
References
Form1.cs
Message.cs
MyGame.cs
Player.cs
Program.cs
MyGameLib
MyGameServer
Program.cs
Source Control Explorer
MyGameClient.Program
namespace MyGameClient
{
static class Program
{
[STAThread]
static void Main(string[] args)
{
Application.EnableVisualStyles();
Application.SetCompatibleTextRe
Application.Run(
new Form1(args[0], args[1], args
}
}
}

When does your code run?

Threads

- **Game Thread**
  - Rendering graphics
  - Respond to mouse/keyboard

```
game = new MyGame(…);
game.Run();
```

- **Windows Forms Thread**
  - Rendering Forms
  - Respond to mouse/keyboard

```
Application.Run(
  new Form1(…));
```

- **QSM Thread (inside QSM)**

```
client = QS.CMS.Framework2.Client.Create(…);
```

  - Network communication
- **Your own threads?**
  - Your background processing etc.

Your Code

- Your own threads
- Callbacks from Windows Forms (GUI Thread)
  - Mouse, keyboard etc.
- Callbacks from Game (Game Thread)
  - **Update(GameTime gameTime)**
  - **Draw(GameTime gameTime)**
- Callbacks from QSM (QSM Thread)
  - When packets are received, sent, acked etc.
  - **Must do minimum work and leave!**

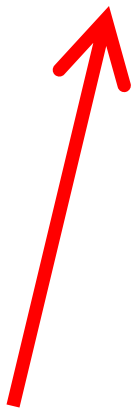
How to use QSM

Launching QSM “client”

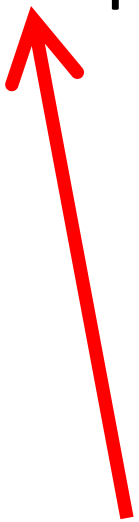
```
using QS.CMS.Framework2;
…
ClientConfiguration configuration =
    new ClientConfiguration(
        “192.168.0.0/24”,
        10000,
        “192.168.0.100”);
configuration.Verbose = true;
…
IClient client = Client.Create(configuration);
```

Joining a QSM group

```
IGroup group = client.Open(
    new QS.CMS.Base3.GroupID(groupid),
    GroupOptions.FastCallbacks);
```



how QSM interacts with your code



numeric identifier

Receiving from a group

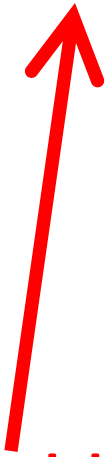
```
group.OnReceive +=
    new IncomingCallback(ReceiveCallback);

private void ReceiveCallback(
    QS.CMS.Base3.InstanceID sender,
    QS.CMS.Base3.ISerializable message)
{
    Message m = (Message) message;
    …
```

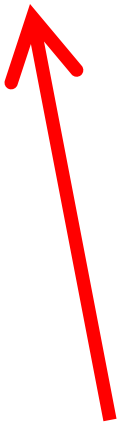
called on receive

your class

Sending to a group (1)

```
group.BufferedChannel.Send(new Message(...));
```

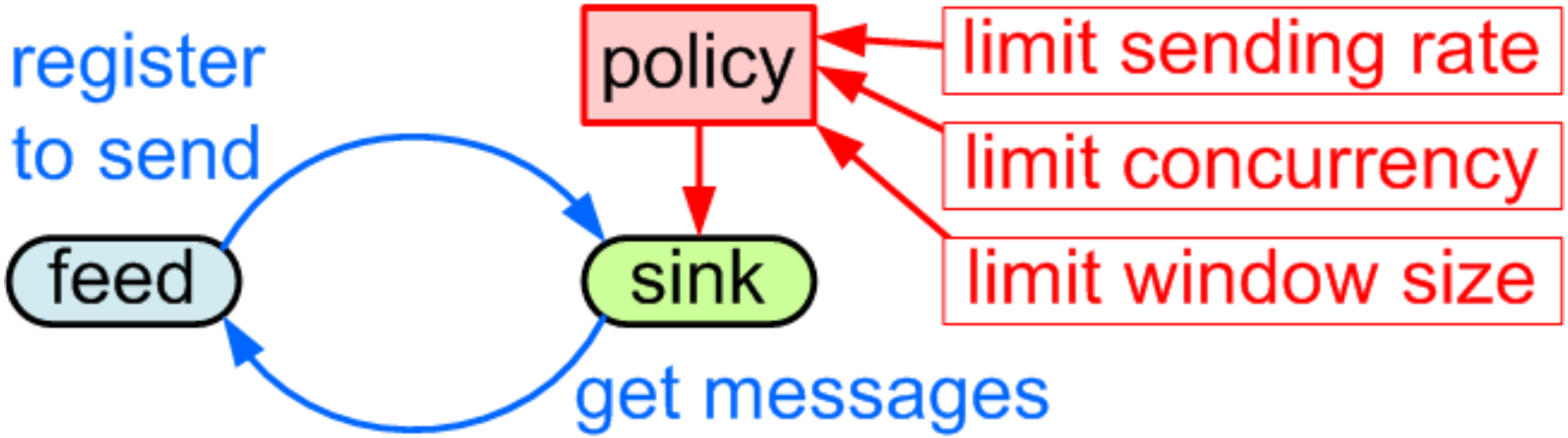


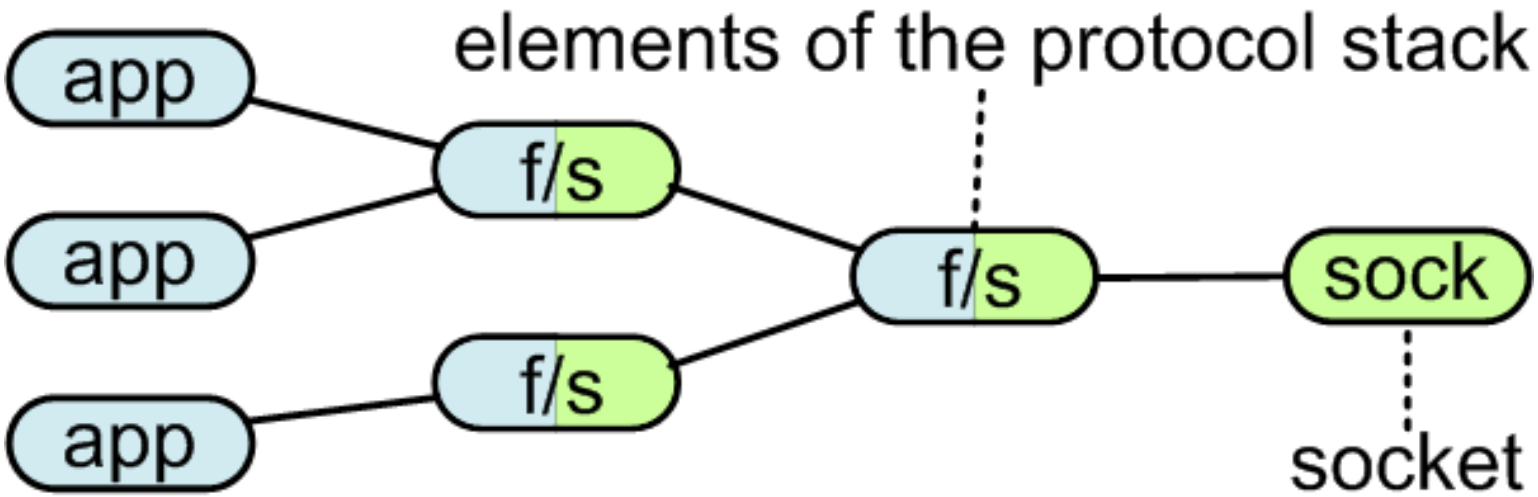


uses internal buffer to store messages you are sending

you can send this way from any thread

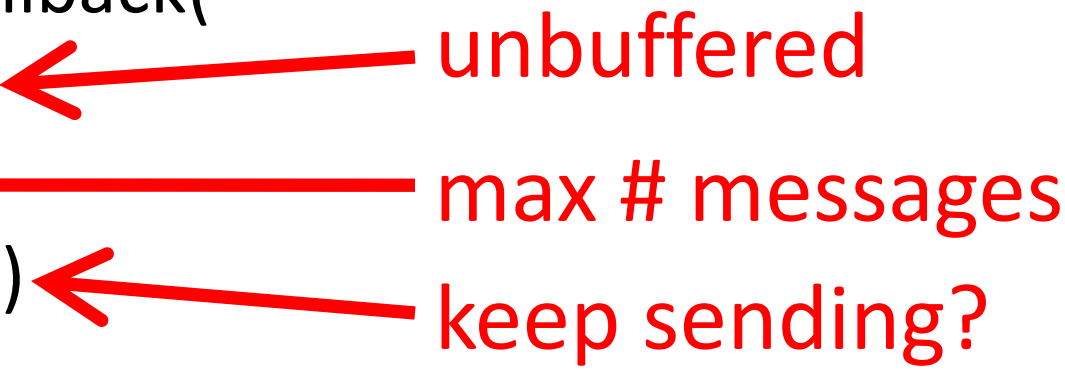
Sending to a group (2a)





Sending to a group (2b)

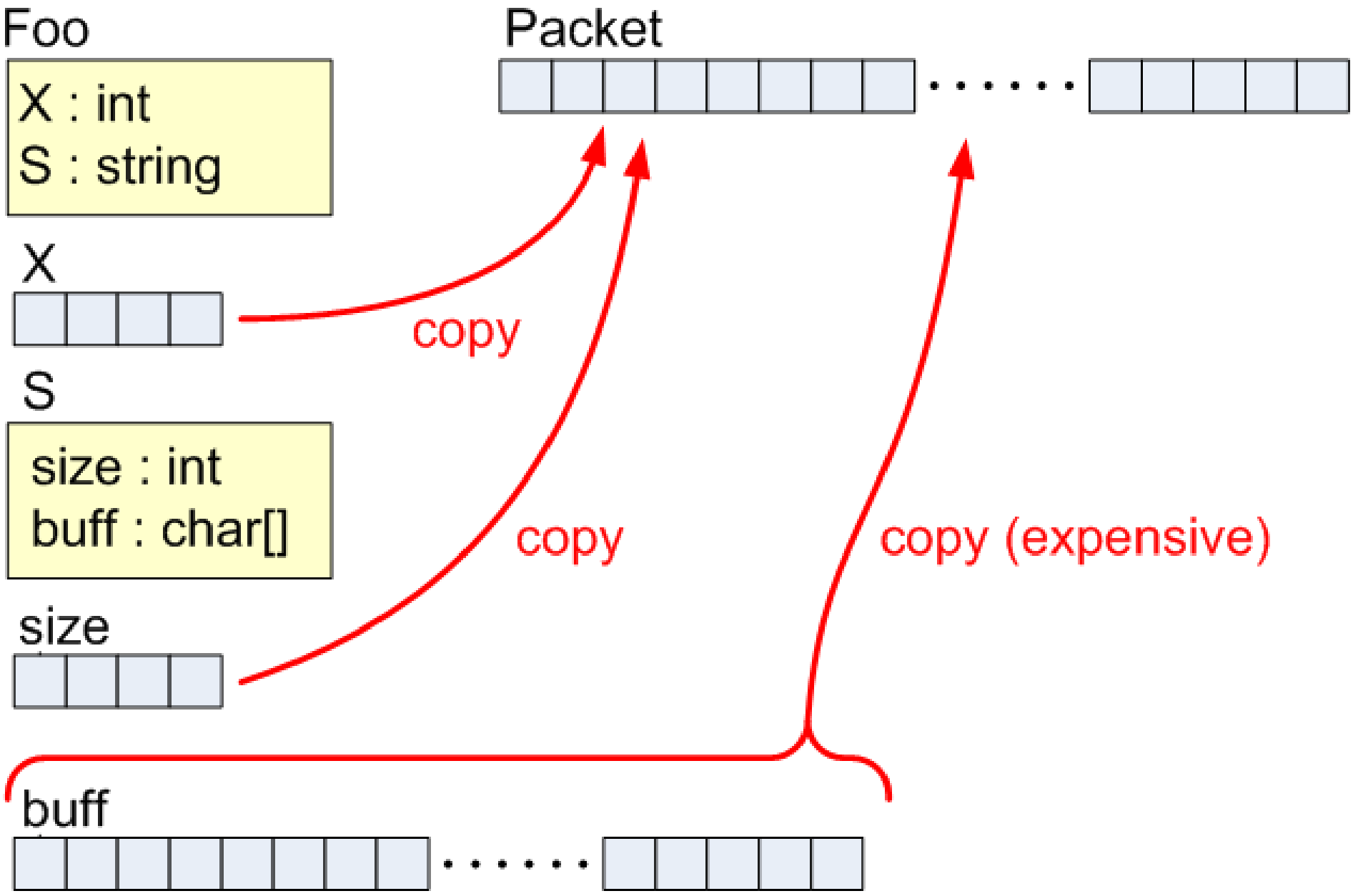
```
group.ScheduleSend(
    new OutgoingCallback(SendCallback));

private void SendCallback(
    IChannel channel,
    uint maxsend,
    out bool hasmore)
{
    channel.Send(new Message(…));
    hasmore = false;
}
```



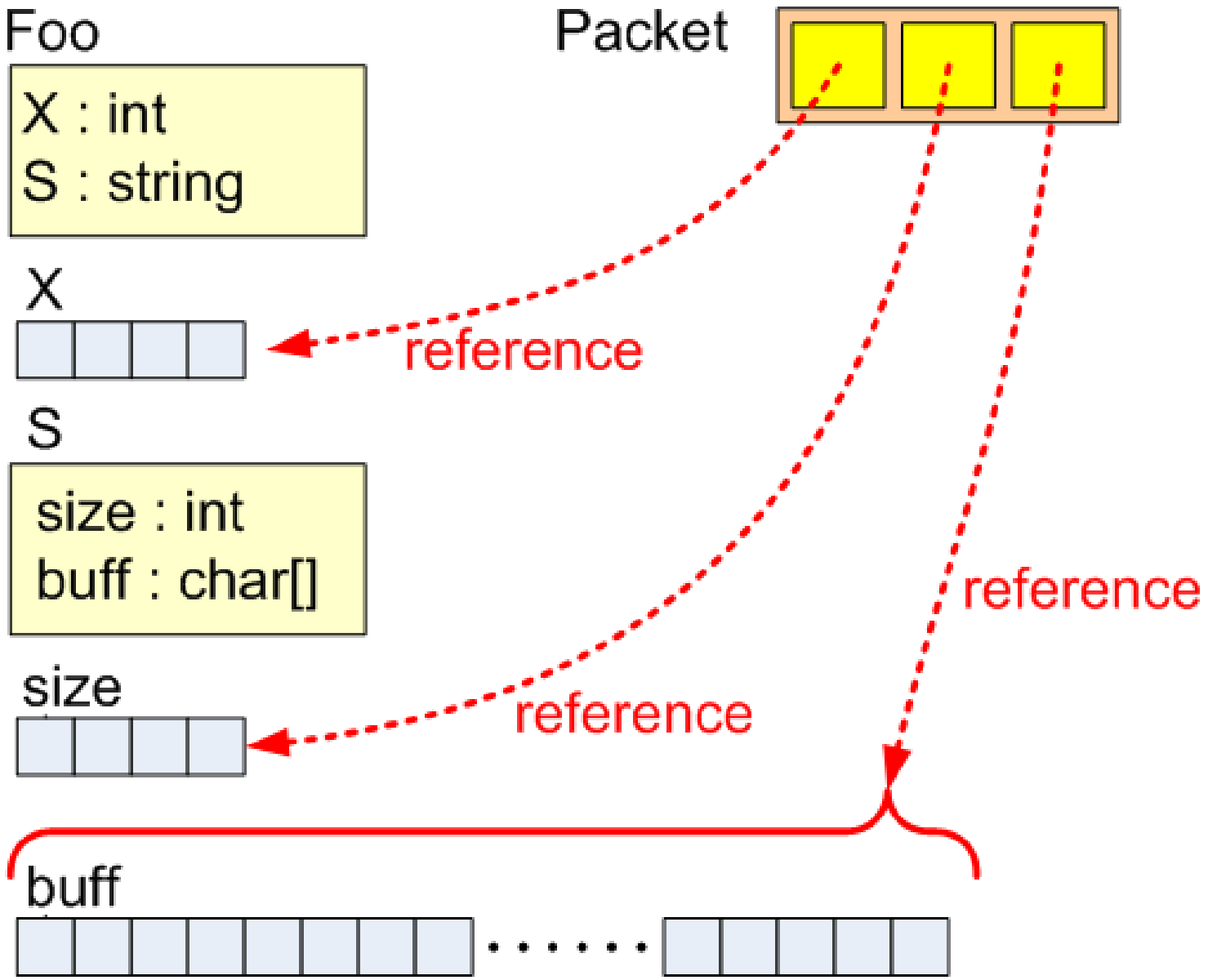
Serialization

- Built-in XML
  - Flexible, easy to use
  - Slow
  - Space-consuming
- Built-in “binary”
  - Useless
- QSM
  - Fast, supports scatter-gather I/O
  - Takes getting used to…

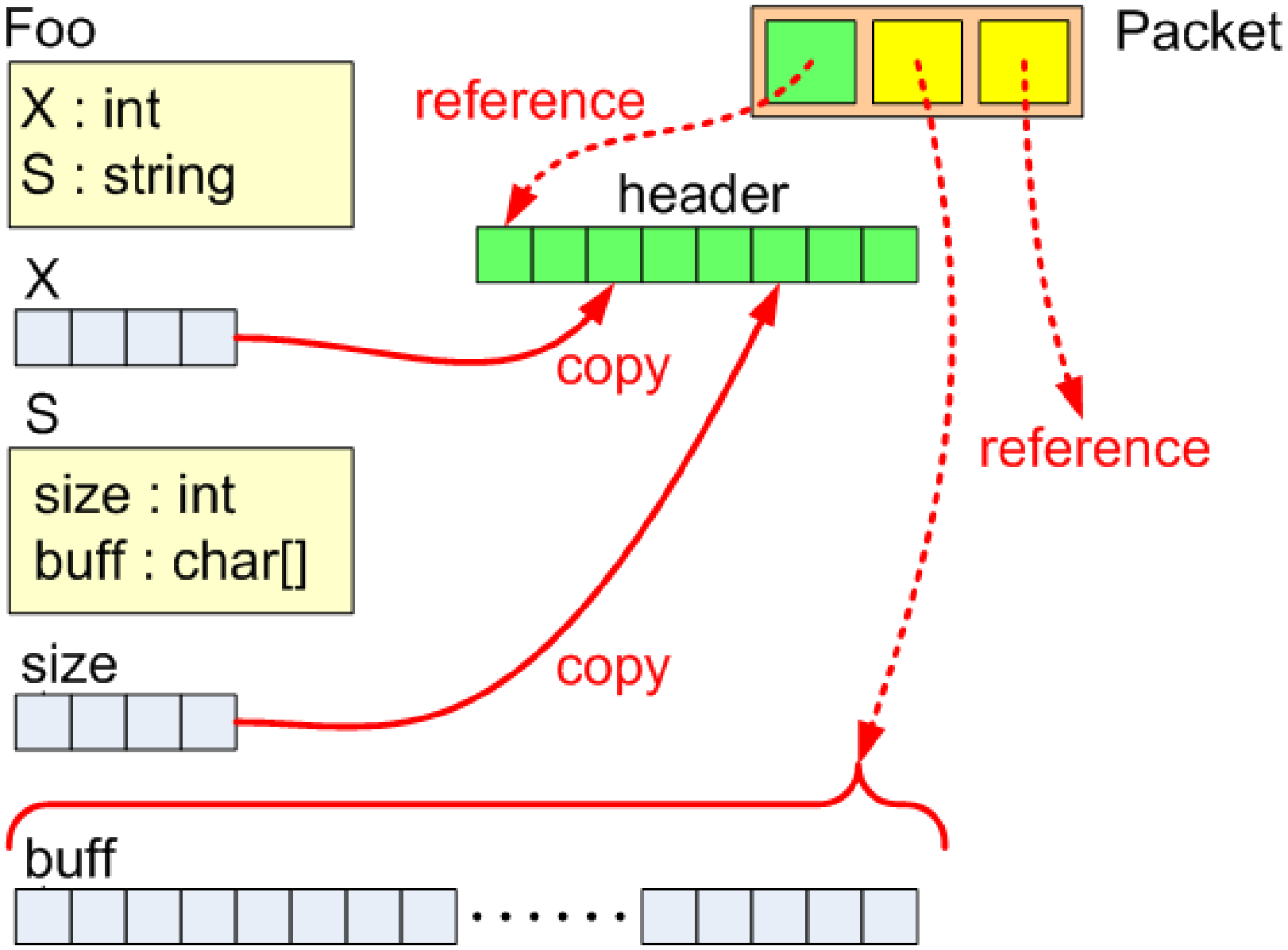
“traditional” serialization



“scatter-gather” serialization



serialization in QSM

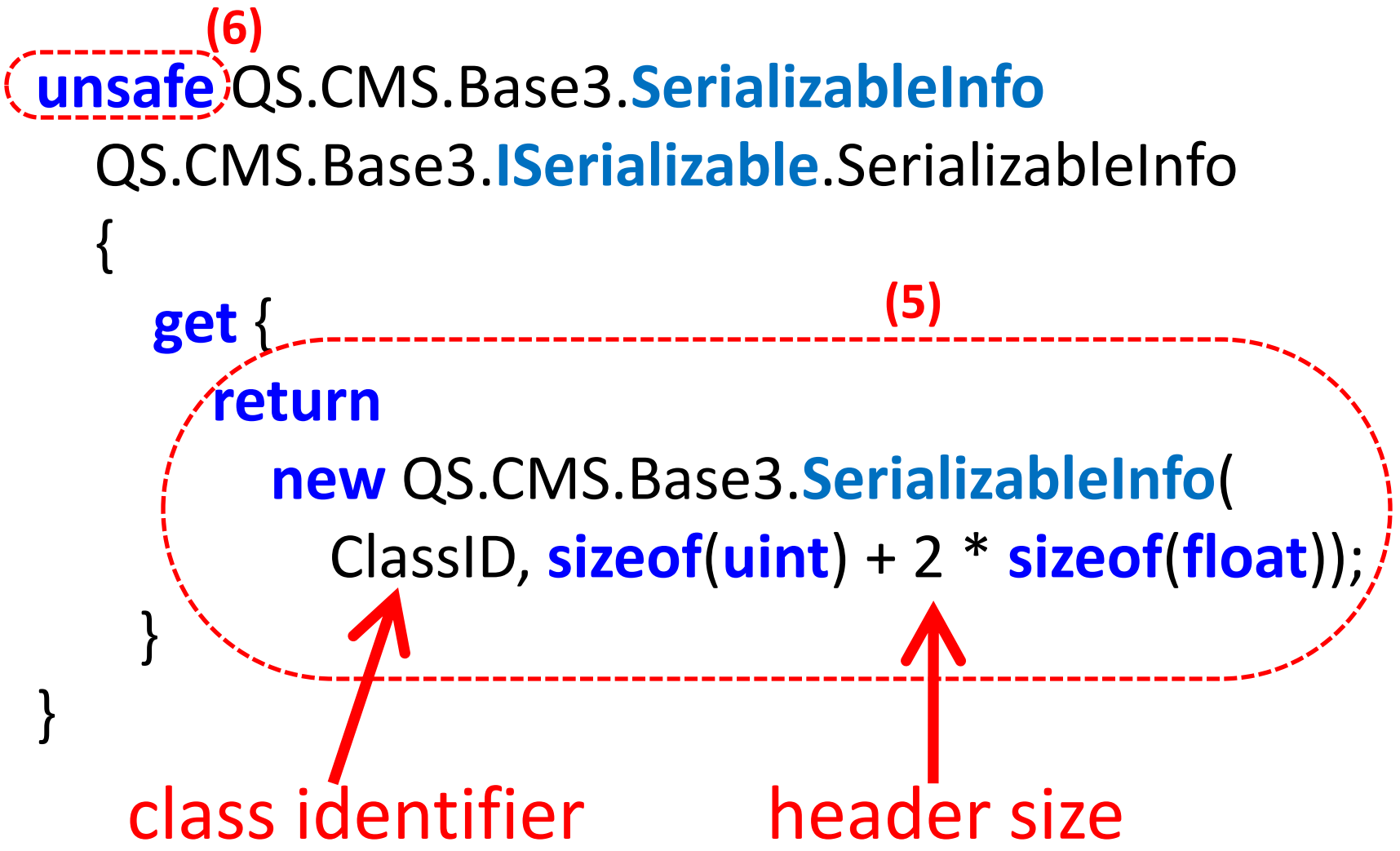


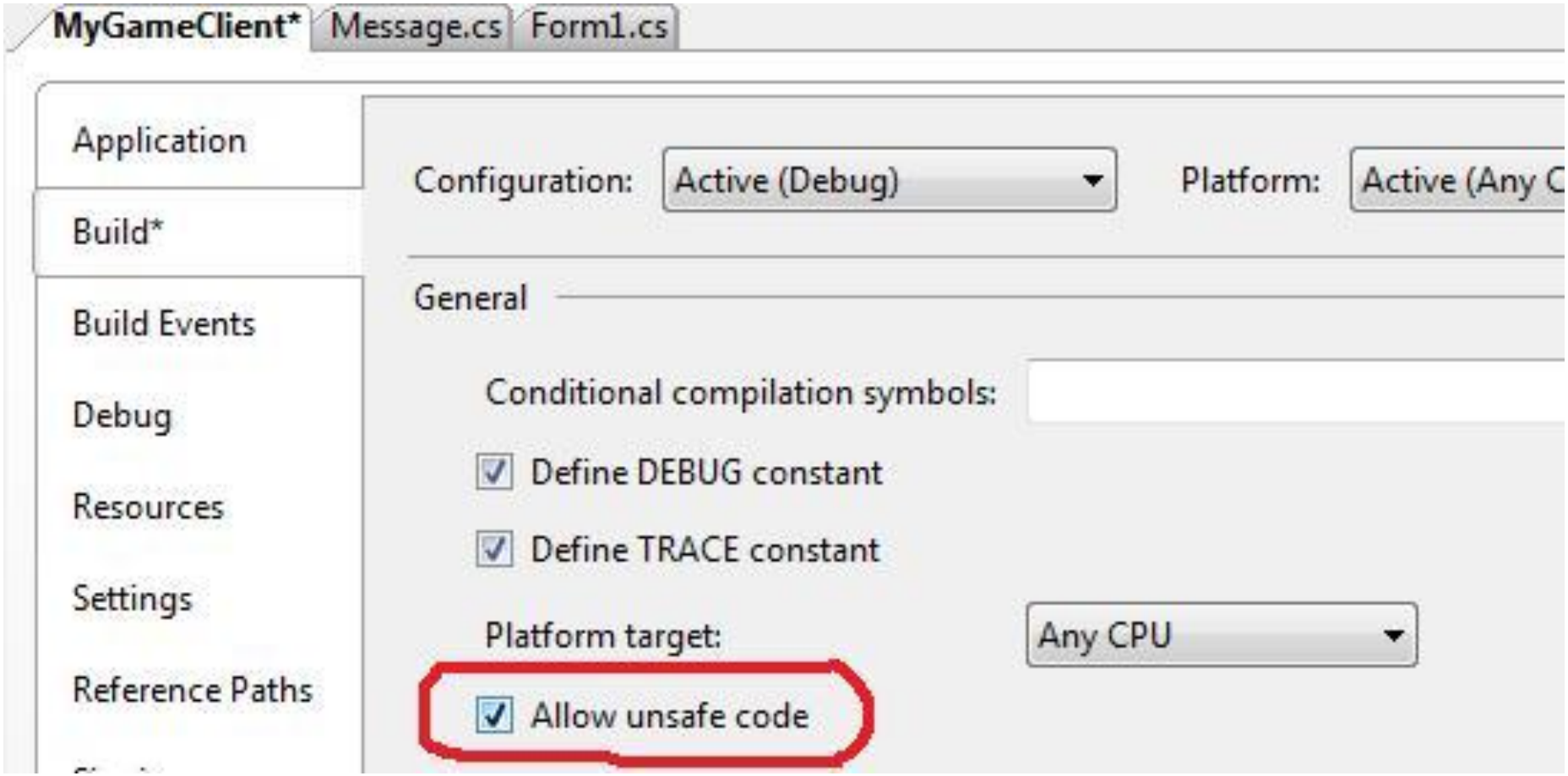
Anatomy of a serializable class (1)

(1)

```
[QS.CMS.Base2.ClassID(Message.ClassID)]                    (3)
public sealed class Message : QS.CMS.Base3.ISerializable
{
    public const ushort ClassID =                          (2)
        (ushort) (QS.ClassID.UserMin + 100);

    public Message()                                       (4)
    {
    }
```

Anatomy of a serializable class (2)




MyGameClient*
Message.cs
Form1.cs
Application
Build*
Build Events
Debug
Resources
Settings
Reference Paths
Configuration:
Active (Debug)
Platform:
Active (Any C
General
Conditional compilation symbols:
Define DEBUG constant
Define TRACE constant
Platform target:
Any CPU
Allow unsafe code

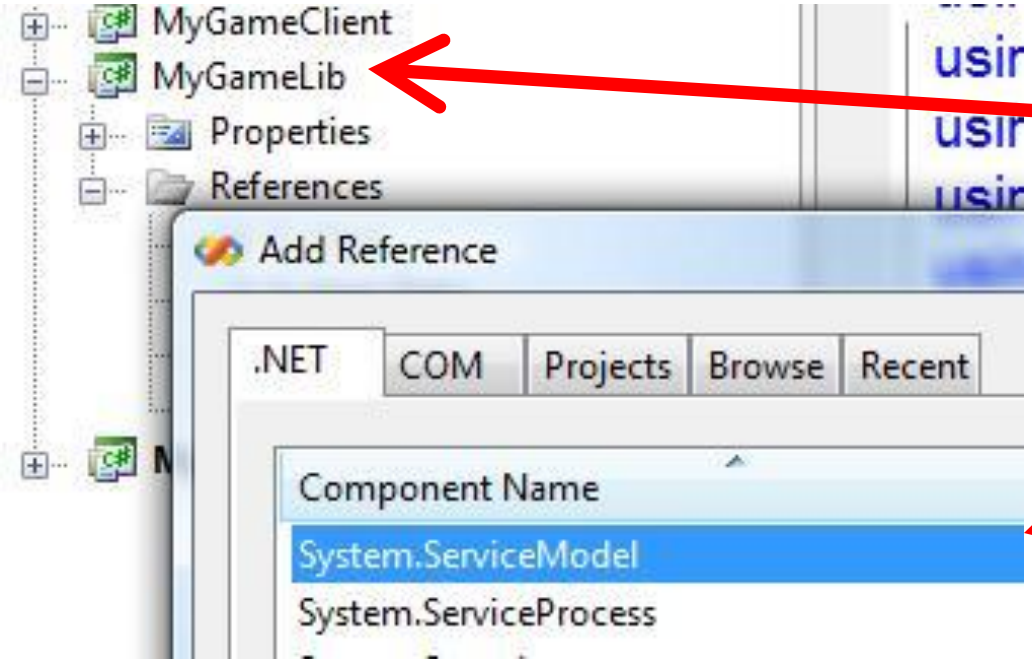
Anatomy of a serializable class (3)

```
unsafe void QS.CMS.Base3.ISerializable.SerializeTo(
    ref QS.Fx.Base.ConsumableBlock header,
    ref IList<QS.Fx.Base.Block> data)
{ (7)
                                    (8)
    fixed (byte* arrayptr = header.Array)
    {
                                                    (9)
        byte* headerptr = arrayptr + header.Offset;
                                                                (10)
        *((uint*)headerptr) = playerid;
        *((float*)(headerptr + sizeof(uint))) = x;
        *((float*)(headerptr + sizeof(uint) + sizeof(float))) = y;
    }
```

here you can copy

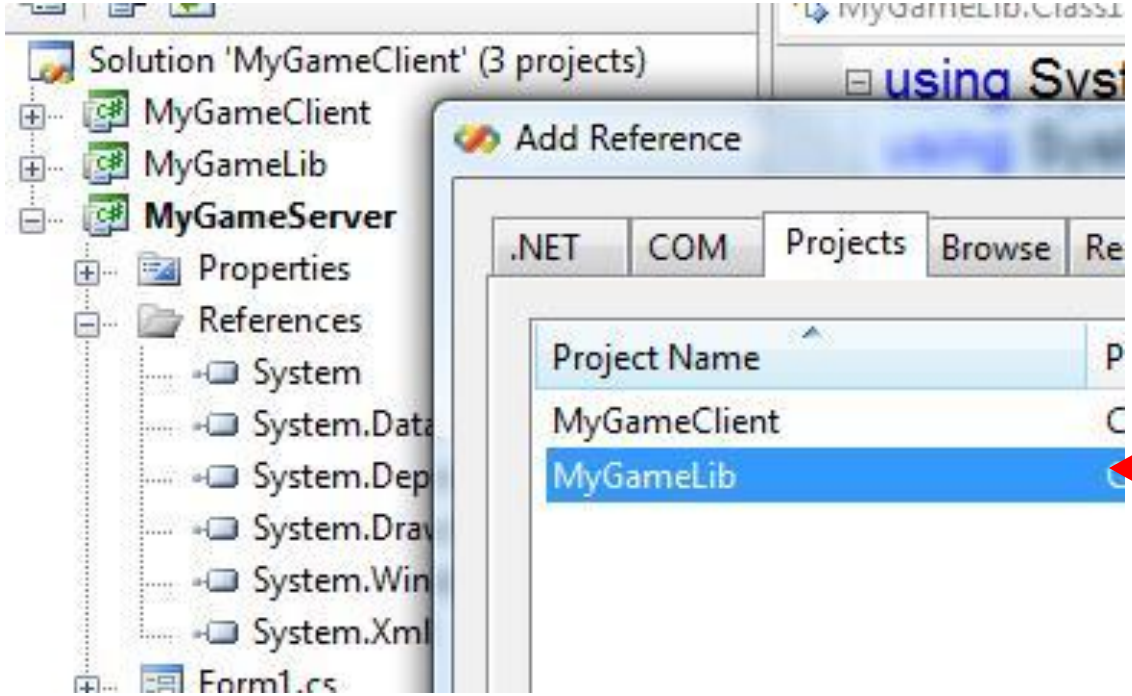
append your buffers here

How to use WCF



shared class library

reference WCF DLLs



reference your class lib from client and server

```
using System.ServiceModel;

namespace MyGameLib {

    [ServiceContract]
    public interface IMyGameServer {

        [OperationContract]
        void JoinTheGame(string myname, byte[] mypicture,
            out uint myid, out uint groupid);

        [OperationContract]
        void GetPlayerInfo(uint playerid,
            out string playername, out byte[] playerpicture);
    }
}
```

```
[ServiceBehavior(
    InstanceContextMode = InstanceContextMode.Single)]
public partial class Form1
    : Form, MyGameLib.IMyGameServer  {
    public Form1(string subnet) {
        …
        servicehost = new ServiceHost(this);
        servicehost.AddServiceEndpoint(
            typeof(MyGameLib.IMyGameServer),
            new WSHttpBinding(),
           "http://" + Dns.GetHostName() + ":60000");
        servicehost.Open();
    }
```

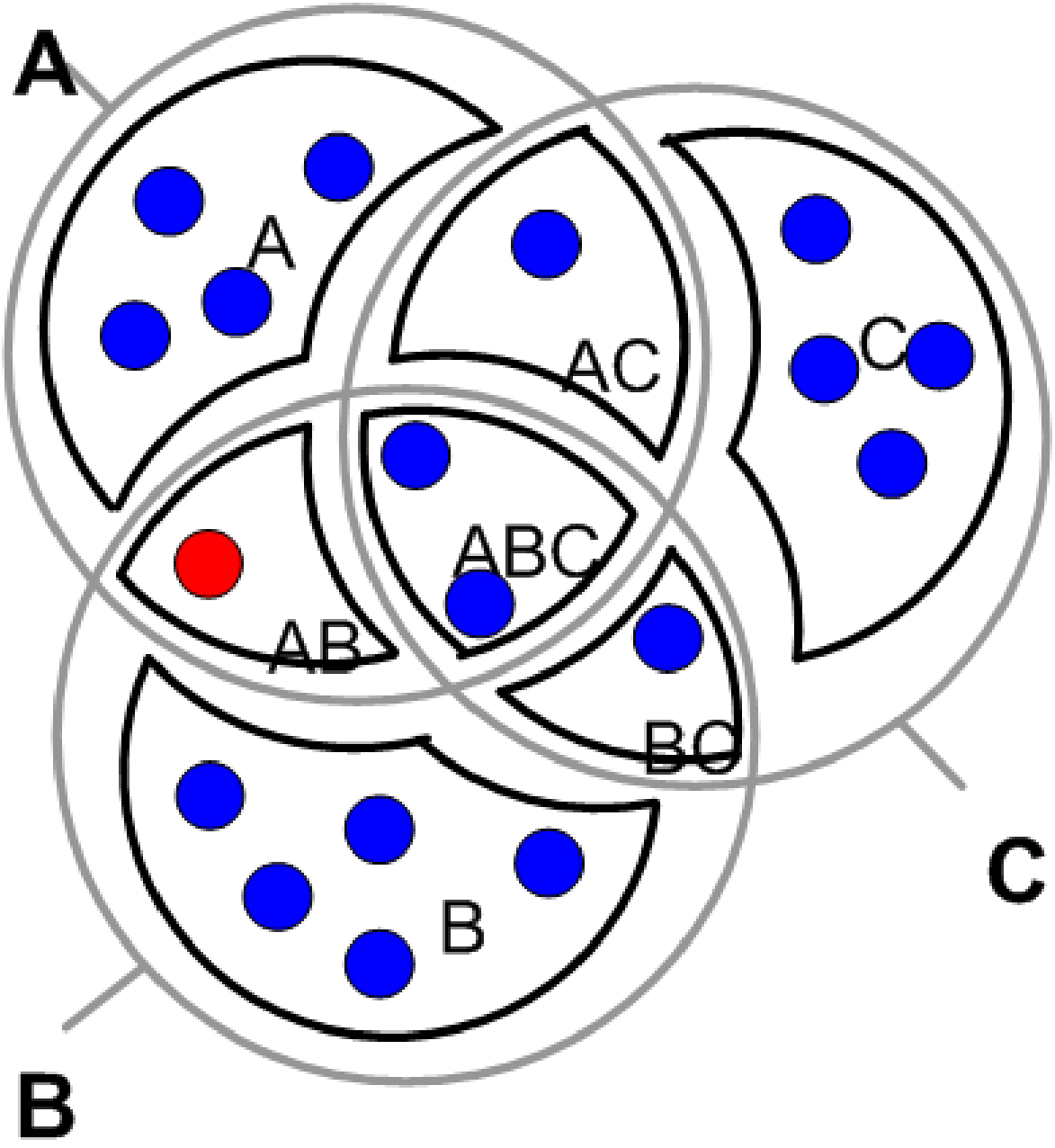
```
ChannelFactory<MyGameLib.IMyGameServer>
    servicefactory =
        new ChannelFactory<MyGameLib.IMyGameServer>(
            new WSHttpBinding(),
            new EndpointAddress(
                "http://" + serveraddress + ":60000"));

IMyGameServer gameserver =
    servicefactory.CreateChannel();

gameserver.JoinTheGame(
    playername, picturebytes, out myid, out groupid);
```

How to use XNA

```
private Player CreatePlayer(
    uint id, string name, byte[] picture,
    Vector2 position)
{
    Player player = new Player(…,
        new SpriteBatch(graphics.GraphicsDevice),
        Texture2D.FromFile(graphics.GraphicsDevice,
            new MemoryStream(picture)),
        …);
    …
}
```

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    foreach (Player player in players.Values)
    {
        player.sprite.Begin(SpriteBlendMode.AlphaBlend);
        player.sprite.Draw(
            player.texture, player.position, Color.White);
        player.sprite.End();
    }
    base.Draw(gameTime);
}
```

How to abuse QSM


A
A
AC
C
ABC
AB
BC
B
C
B

References

Software (1)

- Visual Studio 2005 Professional Edition:
  MSDNAA
- Visual Studio 2005 Express Edition:
  http://msdn.microsoft.com/vstudio/express/
- XNA Framework, XNA Game Studio Express
  http://msdn2.microsoft.com/en-us/xna/
- Other:
  See the tutorial for details (URL on next page)

Software (2)

- QSM:
  - The QUickSilver Project webpage http://www.cs.cornell.edu/projects/quicksilver/QSM/index.htm
  - **Download the latest distribution** (0.12 as of the time of writing this presentation)
  - Report bugs to Krzys
    - krzys@cs.cornell.edu
    - Remember to include logs (see the tutorial for details)

Tutorials (1)

- Video Tutorials on MSDN
  http://msdn2.microsoft.com/en-us/xna/bb245766.aspx
- MSDN Documentation (three examples)

  http://msdn2.microsoft.com/en-us/library/bb203894.aspx
  (you can find this content also in your local help documentation installed with XNA)

Tutorials (2)

- **QSM Tutorial**: http://www.cs.cornell.edu/projects/quicksilver/QSM/tutorial/tutorial.html
- **QSM API**: http://www.cs.cornell.edu/projects/quicksilver/QSM/api/
- **QSM User's Guide**: http://www.cs.cornell.edu/projects/quicksilver/QSM/users_guide.htm

Discussion