

---

# Open Overlay Support for the Divergent Grid

---

Geoff Coulson, **Paul Grace**,  
Gordon Blair, Barry Porter, Wei Cai



Computing Department,  
Lancaster University

David Duce, Chris Cooper, Muhammad  
Younas, Musbah Sagar, Wei Li



Department of Computing,  
Oxford Brookes University

---

# Overview

- The Divergent Grid
  - Gridkit
    - Reflection for dynamic reconfiguration
    - Open Overlays
    - Multi-personality Interaction types
  - Example: “Supporting a Collaborative Workspace”
  - Status & Current Work
-

---

# Introduction

- The Divergent Grid
    - Diversity in terms of end-systems and networked infra-structures:
      - high-speed LANs, lower-speed WANs, infrastructure-based wireless networks, ad-hoc wireless networks
      - cluster systems through to miniature sensor devices.
    - the range of types of “interaction types” in use at the application level has also burgeoned:
      - RPC, Message-Oriented Middleware, Media Streaming, Publish-Subscribe, P2P resource lookup and file sharing, workflow ...
-

---

# The WildFire Scenario

- Forest or Savannah fire fighting
  - Fire-fighters carry PDA devices
    - Communicate with other fire-fighters and controllers
      - Graphic, text and audio commands
      - Wireless networks
  - Deployed sensors
    - Video sensors
    - Portable wind sensors
  - Remotely-located experts
    - Monitor data, visualise the fire based on data
    - Issue commands to firefighters (across network types)
    - Collaborate with one another (fixed network\)
-

---

# Analysing the Scenario

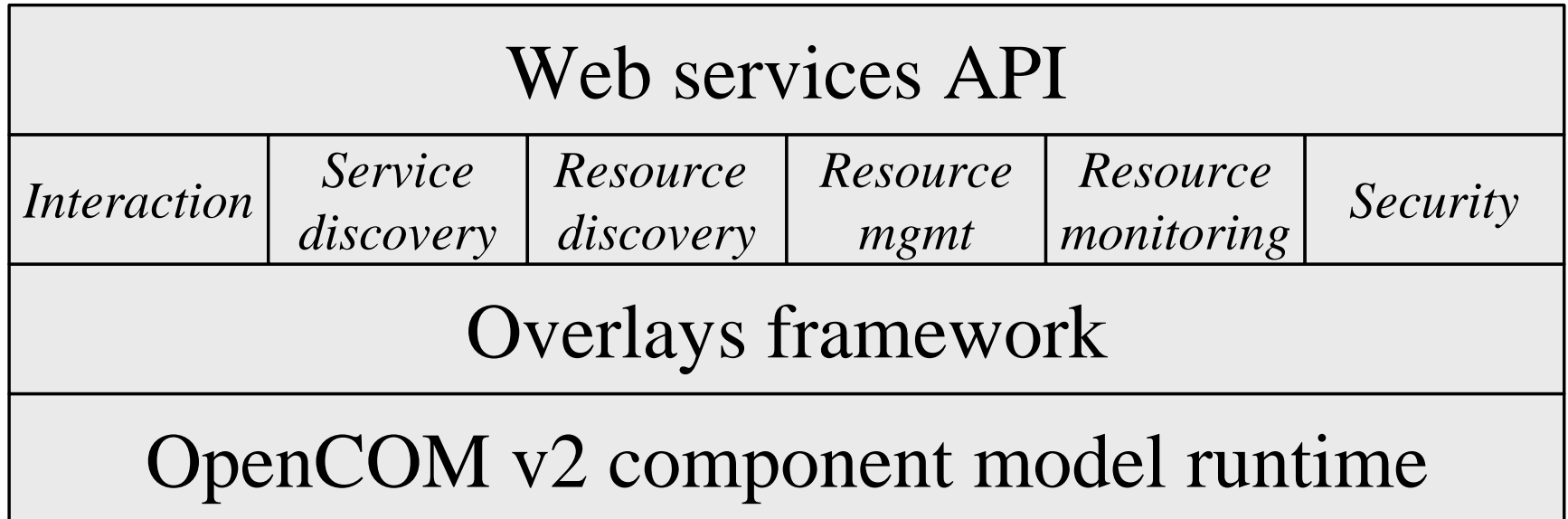
- Highly heterogeneous device and network technologies
    - Sensors, PDAs to high-end workstations
      - *Current Grid middleware is heavyweight, and isn't profilable to work on such a range of devices.*
    - Low-speed WAN, ad-hoc wireless networks
      - *Current Grid middleware is non-network centric, it assumes fixed TCP/IP*
  - Diverse interaction paradigms
    - Reliable group multicast for command propagation
    - Streaming for audio communication
    - Publish-subscribe for sensor events.
      - *Current Grid middleware supports only SOAP messaging*
-

---

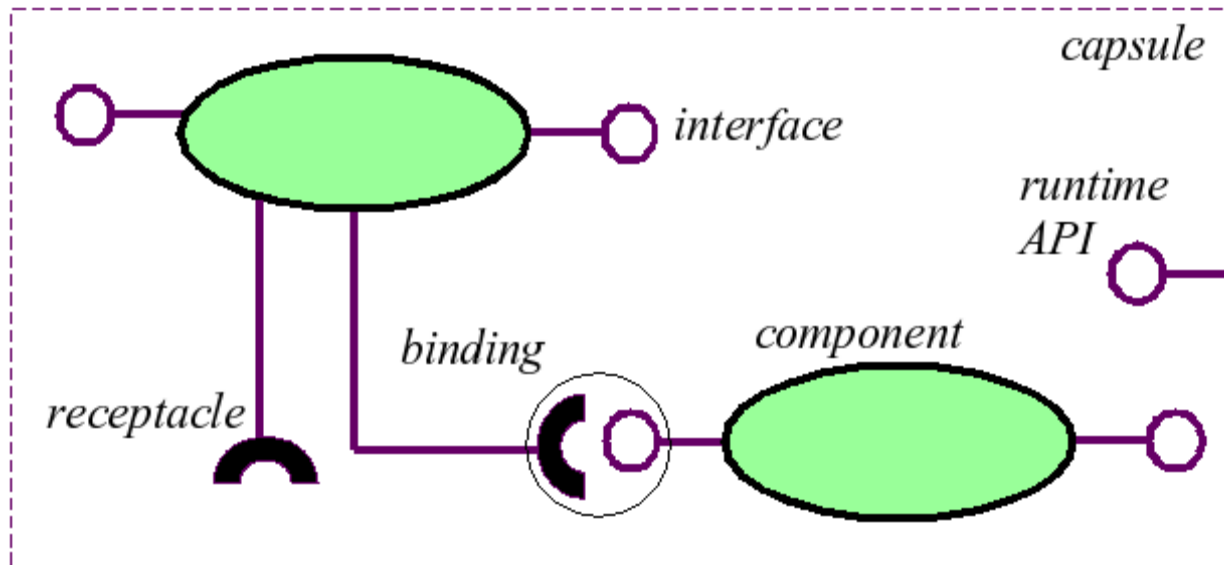
# The Gridkit Approach to Grid Middleware

- Flexible and configurable set of middleware frameworks
    - Over a layer of Overlay networks
    - Lightweight component model implemented upon a wide range of device types
  - Benefits of the Overlay approach
    - i) hide the inherent network heterogeneity
    - ii) implement network services not supported in the network
    - iii) highly configurable and run-time adaptive
-

# The Overall Gridkit Architecture



# The OpenCOM Component Model



Available at:

<http://www.comp.lancs.ac.uk/computing/research/mpg/reflection/opencom.php>



---

# The Reflective Middleware Approach

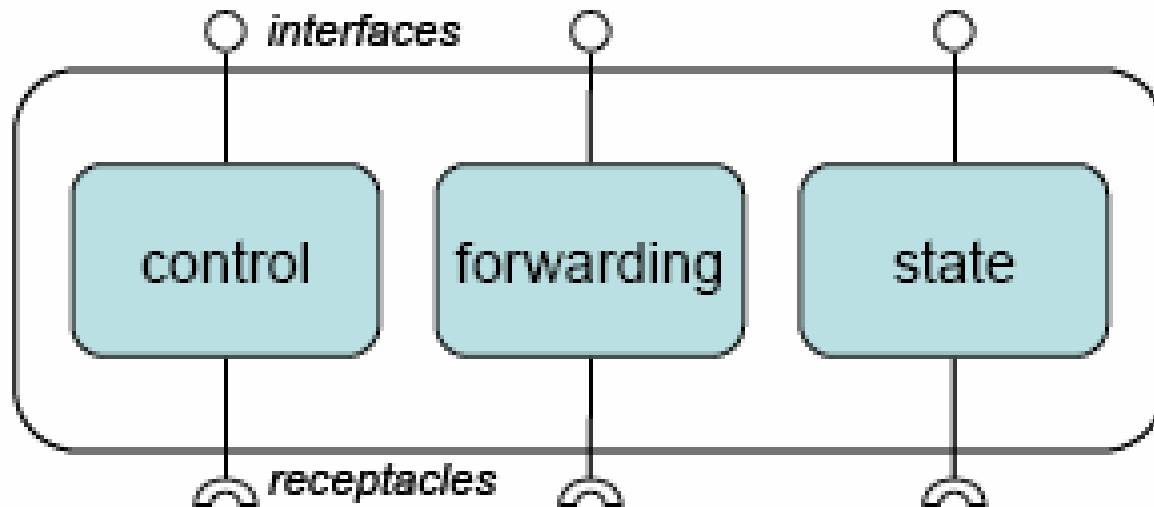
- Components
    - Middleware building blocks
    - Support re-use, configurability, and reconfigurability
  - Reflection
    - Introspection
      - Component architecture, and system behaviour
    - Runtime adaptation
      - Dynamically change structure & behaviour
  - Component Frameworks
    - Manage domains of middleware functionality
    - Maintain system integrity
-

---

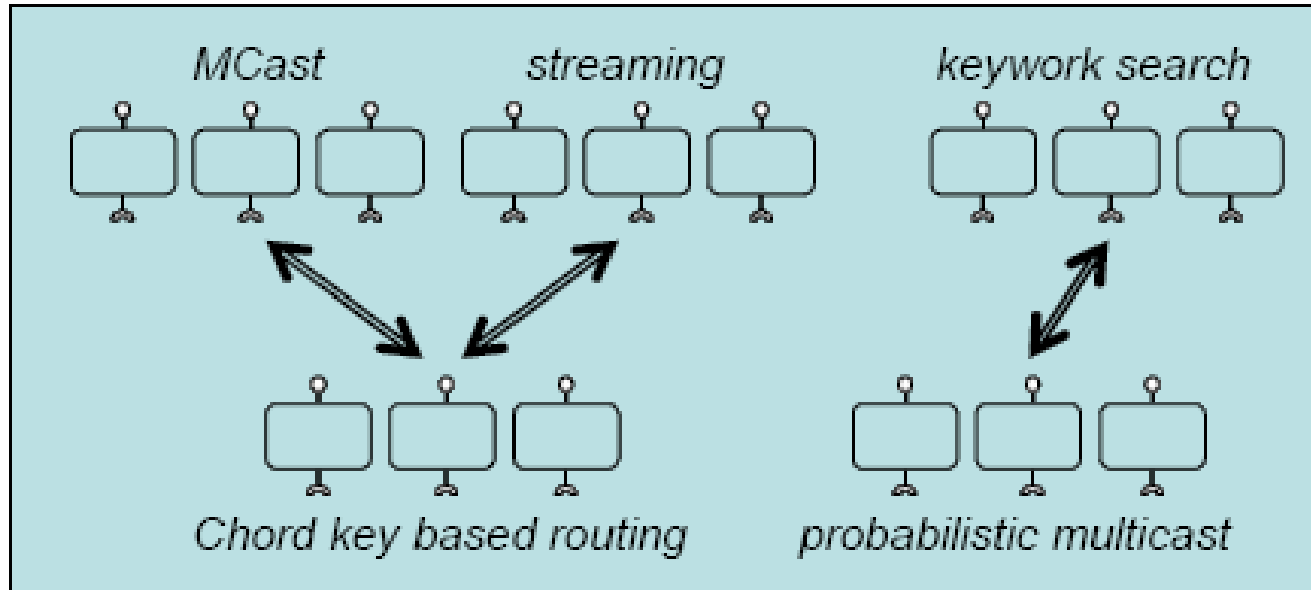
# Overlay Networks

- Virtual Communication structures logically laid over physical network
    - Lots of types – structured DHT based rings, large-scale multicast trees, unstructured keyword search, overlays for ad-hoc networks .. and so on..
  - Hot topic in distributed systems research
    - adding network services above overlays
      - Pastry team– Scribe, PAST, POST, Structella etc..
      - Common APIs for overlays for re-use.
      - This approach fits well with our middleware framework approach
  - We propose more general approach
    - Overlays depend arbitrarily (sensibly!) on other overlays
    - Overlays implemented to a common plug-in format
-

# Structure of an Overlay Plug-in

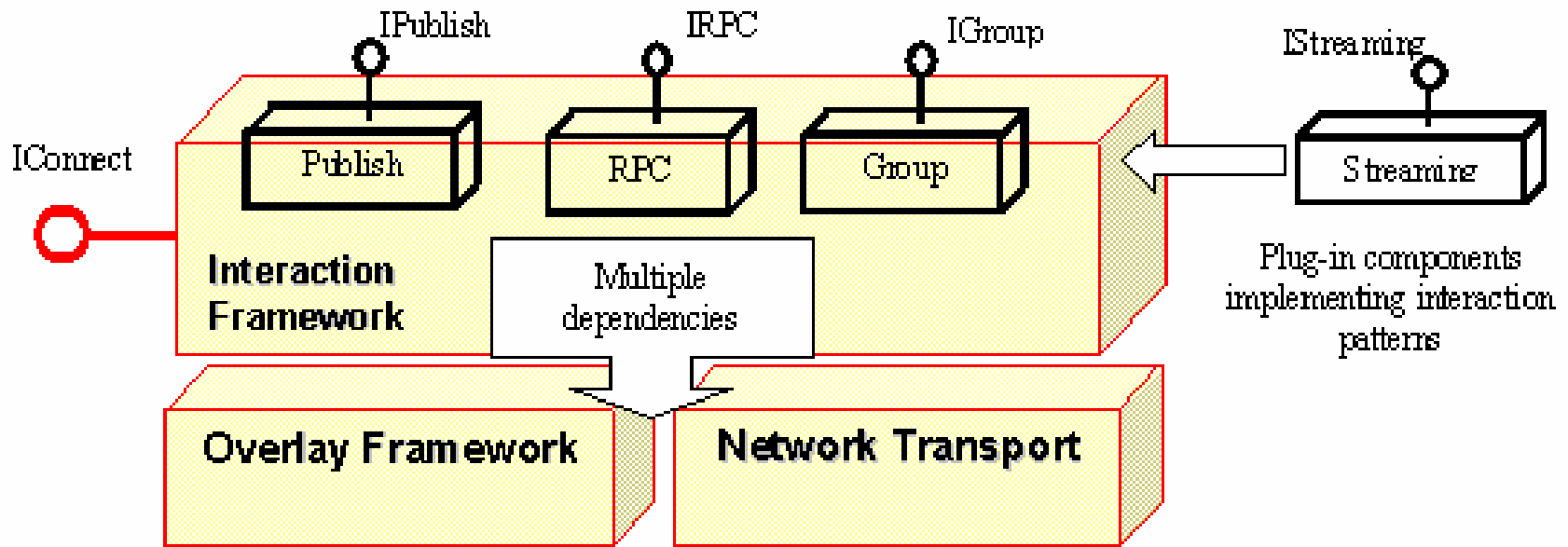


# Example Overlay Configurations



# The Interaction Framework

- Trading scheme for suitable interaction
  - Attach declarative requirements to the application receptacle and connect to *IConnect*
  - *Context-based*
  - *Contract monitoring*



# Trading for Middleware Behaviour

Framework	Generic API	Item	Name-value pairs
Interaction	<i>IPublish</i>	Publish	RelMes: F
	<i>IGroup</i>	Group1	RelMes: F; GrpMem: T
		Group2	RelMes: F; GrpMem: F
Overlay	<i>IGroupMessage</i>	ALM	RelMes: F; Net: fixed
	<i>IGroupMessage</i>	ProbMcast	RelMes: F; Net: adhoc
	<i>IGroupMembers</i>	Gossip	RelMes: F; Net: fixed; Net: adhoc
Context	N/A	PC	Net: fixed
		PDA	Net: adhoc

# Implementing the WildFire Scenario

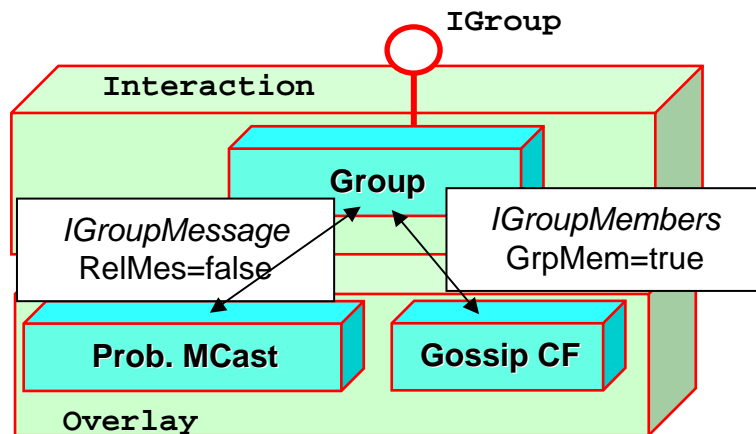
The image shows a web browser window displaying an SVG map of a wildfire scenario. The browser's address bar shows the file path: `C:\Desktop\svgCWE\svgCWE.svg`. The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help) and a toolbar with navigation and search icons.

The main content area displays a satellite-style map with a red oval highlighting a specific area. A white arrow points from this oval to a red oval in the XML browser window. Another white arrow points from the red oval in the XML browser to a red oval in the map. A stick figure is visible on the map, and a red square is also present. The XML browser window shows the following code:

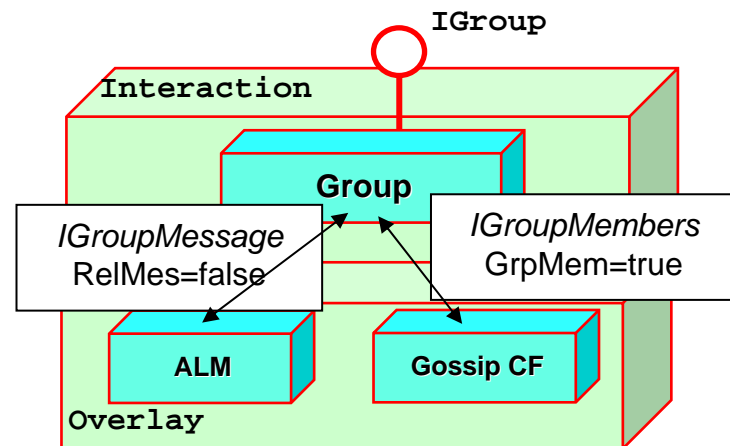
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <metadata id="metadata">
    <rdf:RDF>
      <geom2d:SVGPath rdf:about="User[guest]:PolyLineFigure@4">
        <geom2d:pathData> M 899 519 L 868 517 L 873 515 L 877 515 L 881 513 L 885
512 L 889 511 L 893 510 L 896 508 L 900 507 L 904 504 L 908 501 L 912 498 L 915
495 L 918 492 L 920 488 L 924 486 L 926 482 L 928 477 L 930 474 L 934 469 L 935
465 L 937 460 L 938 456 L 939 453 L 940 450 L 940 447 L 941 443 L 942 440 L 943
437 L 944 434 L 945 431 L 945 428 L 945 425 L 946 421 L 946 416 L 946 412 L 946
409 L 947 406 L 947 402 L 947 399 L 947 394 L 947 390 L 945 387 L 944 384 L 943
381 L 942 379 L 941 376 L 940 374 L 938 372 L 936 370 L 936 366 L 935 366 L 933
364 L 931 362 L 928 360 L 925 358 L 923 358 L 921 357 L 918 355 L 916 355 L 913
353 L 910 352 L 908 351 L 906 351 L 903 349 L 900 348 L 897 347 L 894 347 L 891
347 L 888 345 L 886 345 L 884 345 L 883 345 L 882 345 L 881 345 L 880 345 L 879
345 L 877 345 L 875 346 L 873 347 L 872 348 L 870 349 L 869 350 L 868 351 L 867
352 L 866 352 L 865 353 L 863 354 L 862 355 L 860 357 L 858 359 L 857 360 L 856
361 L 855 361 L 854 361 L 854 362 L 853 363 L 852 364 L 850 366 L 849 366 L 849
367 L 848 368 L 847 369 L 847 370 L 846 371 L 845 373 L 844 375 L 843 377 L 842
378 L 841 379 L 840 380 L 839 381 L 838 382 L 838 383 L 837 384 L 836 385 L 836
386 L 836 388 L 835 388 L 834 389 L 834 390 L 833 392 L 833 394 L 832 394 L 831
395 L 831 396 L 830 397 L 830 398 L 830 399 L 830 397 L 830 395 L 830 393 L 829
391 L 829 389 L 829 388 L 829 386 L 828 384 L 828 382 L 827 380 L 827 378 L 827
378 L 827 376 L 827 374 L 827 373 L 827 372 L 827 371 L 827 372 L 827 374 L 827
376 L 827 378 L 827 379 L 827 381 L 827 383 L 827 385 L 827 387 L 828 388 L 829
391 L 829 392 L 829 394 L 829 395 L 829 396 L 829 397 L 829 398 L 830 398 L 831
398 L 832 398 L 833 398 L 833 397 L 834 396 L 836 396 L 837 394 L 839 393 L 841
393 L 841 392 L 843 392 L 844 391 L 845 391 L 846 390 L 847 390 L 848 389 L 848
389 L 850 388 </geom2d:pathData>
      <geom2d:SVGPath>
        <geom2d:Ellipse rdf:about="User[guest]:EllipseFigure@1">
          <geom2d:cx>453</geom2d:cx>
          <geom2d:cy>388</geom2d:cy>
          <geom2d:rx>74</geom2d:rx>
          <geom2d:ry>68</geom2d:ry>
        </geom2d:Ellipse>
        <geom2d:Rect rdf:about="User[guest]:RoundRectangleFigure@2">
          <geom2d:cx>814</geom2d:cx>
          <geom2d:cy>54</geom2d:cy>
          <geom2d:width>94</geom2d:width>
          <geom2d:height>124</geom2d:height>
          <geom2d:rx>27</geom2d:rx>
          <geom2d:ry>33</geom2d:ry>
        </geom2d:Rect>
      <geom2d:SVGPath rdf:about="User[guest]:PolyLineFigure@0">
        <geom2d:pathData> M 893 442 L 869 446 L 864 446 L 858 446 L 852 446 L 846
448 L 842 446 L 838 446 L 833 447 L 829 447 L 826 448 L 822 448 L 817 449 L 813
449 L 809 450 L 805 450 L 800 451 L 796 451 L 792 452 L 787 452 L 783 453 L 779
453 L 774 454 L 770 454 L 766 455 L 762 455 L 757 456 L 753 456 L 750 457 L 746
457 L 743 457 L 739 458 L 735 459 L 731 459 L 726 460 L 723 460 L 721 460 L 718
461 L 715 461 L 710 461 L 706 462 L 703 462 L 699 463 L 694 463 L 690 464 L 686
```

# Generated Configurations

- Application need not be concerned by heterogeneity
  - Same application used by fire-fighter & expert
  - Common “contract” – IGroup with GMS and unreliable messaging
  - Gridkit generates the appropriate configuration for the context (device, network)



PDA Configuration



PC Configuration



---

# Status & Current Work

- Large body of implementation in both Java and C++ for PDAs and PCs
    - Pub-Sub, Group, RPC (SOAP, IIOP), Streaming
    - Chord, Scribe, TBCP (ALM), Prob. MCast
    - To appear: <http://sourceforge.net/projects/gridkit>
  - Current work
    - Sensors (Berkeley Motes, Contiki OS)
    - Autonomic reconfiguration of deployed middleware and overlays
    - e-Environment project with leading environmental scientists (e.g. LEC, Proudman Oceanographic library, CCLRC) to apply Gridkit to flood forecasting and water quality control
-