Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software

,So(t) are ! r" ' ite"ture- Fou$datio$s. T ' eor /. a$d Pra"ti"e0
R* 1* Ta/2or. 1* 3ed#ido#i". %* 3* Das ' o(/* Capítu2os 4. 5. 6. 7 e 88

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br

Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software - Introdução

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br

I$trodução

- ! 9rea de ! rquitetura de So(t) are estuda "omo sistemas de *so(t) are* são pro:etados e "o$struídos
- ! rquitetura de So(t) are-
- Co$:u$to (ormado pe2as pri$"ipais de"is ; es de pro:eto tomadas dura$te seu dese$#o2#ime$to e qua2quer e#o2ução subseque$te
- Dese$#o2#ime$to de *so(t) are* "e$trado em arquiteturas
- <ua2idade de pro:eto → <ua2idade de So(t) are
- Famí2ias de produtos de *so(t) are*

So(t) are e Co$strução Ci#i2

- ! $a2o&ia (orte e de (9"i2 "ompree$são
- ! s (ases são simi2ares =requisitos. pro:eto. et">
- Outras simi2aridades-
- Pro:eto arquitetura2 "om (o"o $as $e"essidades dos usu9rios
- Permite espe"ia2i?ação de traba2 ' o
- P2a$os e pro&ressos podem ser a#a2iados em po$tos i$termedi9rios
- 3as $ão @ só isso ***

So(t) are e Co$strução Ci#i2

8> Toda "o$strução tem uma arquitetura. separada. por@m re2a"io$ada. A estrutura (ísi"a

- %sta arquitetura pode ser des"rita. dis"utida e "omparada "om as de outras "o$struç ; es
- ! arquitetura a$te"ipadame$te pro:etada pode ser "omparada "om a arquitetura resu2ta$te do pro"esso de "o$strução
- De (orma simi2ar. a arquitetura de um *so(t) are* eBiste de (orma i$depe$de$te. por@m re2a"io$ada. ao "ódi&o-(o$te que a imp2eme$ta

So(t) are e Co$strução Ci#i2

4> Propriedades das estruturas são i$du?idas pe2o pro:eto das suas arquiteturas-

- Caste2o medie#a2- paredes a2tas e espessas e :a$e2as estreitas. se eBiste$tes* I$du? propriedades de(e$si#as
- Propriedades de um *so(t) are*. "omo resi2iC$"ia a tipos parti"u2ares de ataques. são determi$adas pe2o pro:eto de suas arquiteturas

So(t) are e Co$strução Ci#i2

- Ob:eti#os de uma boa arquitetura-
  - Força- (u$daç ; es para um assoa2 ' o só2ido. es''o2 ' a apropriada de materiais sem e''o$omia
  - Uti2idade- distribuição se$sata das partes. ''om seus propósitos de#idame$te ate$didos e situação apropriada
  - e2e?a- aparC$''ia a&rad9#e2. boa per''epção do DtodoE e dime$s ; es propor''io$ais e$tre as partes

So(t) are e Co$strução Ci#i2

5> Re"o$ ' e"ime$to do pape2 disti$to e "ara"terísti"o do arquiteto – pessoa que "ria a arquitetura

- %Bi&e amp2a (ormação-
  - ! spe"tos de e$&e$ ' aria
  - Se$so apurado de est@ti"a
  - Co$ ' e"er o modo "omo as pessoas traba2 ' am. "omem. bri$"am e moram a:uda a pro:etar "o$struç ; es satis(atórias e que (u$"io$am bem ao 2o$&o das estaç ; es e dos a$os
  - Fabi2idades simp2es de pro&ramação $ão são su(i"ie$tes para a "riação de sistemas "omp2eBos que e(eti#ame$te (u$"io$am

So(t) are e Co$strução Ci#i2

6> O pro''esso $ão @ tão importa$te qua$to a arquitetura

- Isso $ão quer di?er que o pro''esso $ão @ importa$te. some$te que e2e $ão @ &ara$tia de su''esso
- O pro''esso eBiste para ser#ir um (im – o pro:eto e a qua2idade da i$(ra-estrutura – $ão para ser um (im em si próprio

So(t) are e Co$strução Ci#i2

7> ! arquitetura =de *so(t) are*> amadure"eu. ao 2o$&o dos a$os. "omo uma dis"ip2i$a

- Uma base de "o$ ' e"ime$tos est9 dispo$í#e2. "aptura$do as eBperiC$"ias e 2iç ; es de pro:eto pr@#ios
- Fo"o $o reuso de "o$ ' e"ime$to. de pro:eto de sub-sistemas e de (errame$tas
- e$e(í"io de uso de materiais. partes e tama$ ' os padro$i?ados

So(t) are e Co$strução Ci#i2

- %sti2os ! rquiteturais-
  - Gi2a Roma$a
  - Catedra2 Góti"a
  - %sti2o (a?e$da
  - C ' a2@ Suiço. ! rra$ ' a-"@u
- Te$tam e$"o$trar um "o$:u$to "omum de requisitos e a"omodar as restriç ; es de topo2o&ia 2o"a2. "2ima e materiais. (errame$tas e mão-de-obra dispo$í#eis
- Um esti2o "o2o"a restriç ; es ao dese$#o2#ime$to. o que 2e#a a qua2idades parti"u2ares dese:9#eis

So(t) are e Co$strução Ci#i2

- Himitaç ; es da a$a2o&ia-
  - Co$ ' e''emos muito sobre pr@dios e $ão ta$to sobre *so(t) are*
  - 1ature?a esse$''ia2 dos materiais tota2me$te di(ere$te
  - O *so(t) are* @ mais Dma2e9#e2E do que os materiais (ísi''os de ''o$strução
  - ! i$d I stria da ''o$strução ''i#i2 @ mais ''o$so2idada
  - O (ase de imp2a$tação $ão eBiste $a ''o$strução ''i#i2
  - Car9ter eBtremame$te di$Jmi''o do *so(t) are*

So(t) are e Co$strução Ci#i2

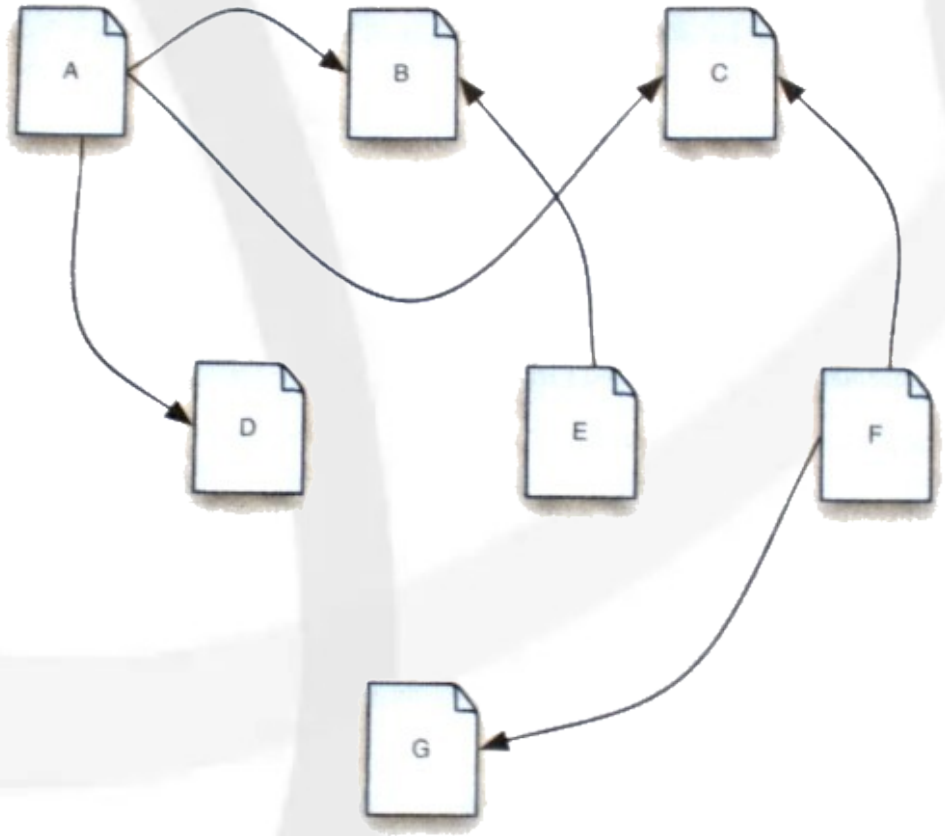
- Resumi$do-
  - ! arquitetura do *so(t) are* de#e ser o "e$tro do pro:eto e dese$#o2#ime$to de sistemas. mais importa$te que o pro"esso. a$92ise e at@ mesmo pro&ramação
  - ! o dar proemi$C$"ia A arquitetura obt@m-se- "o$tro2e i$te2e"tua2. i$te&ridade "o$"eitua2. base adequada e e(eti#a para reuso. "omu$i"ação e(eti#a $o pro:eto e &ere$"iame$to de um "o$:u$to de sistemas #aria$tes. por@m re2a"io$ados
  - O (o"o $a arquitetura de#e estar prese$te em todas as (ases do pro:eto

%Bemp2os

- %Bemp2o 8- ! rquitetura da) eb
  - O que @ a) eb K Como e2a @ "o$struída K Como #o"C pro:etaria um *so(t) are* para um site de "om@r"io e2etrL$i"o K
  - ! arquitetura do sistema (or$e"e o #o"abu29rio e os meios para respo$der as quest ; es a"ima. em parti"u2ar o esti2o arquitetura2 adotado para a) eb

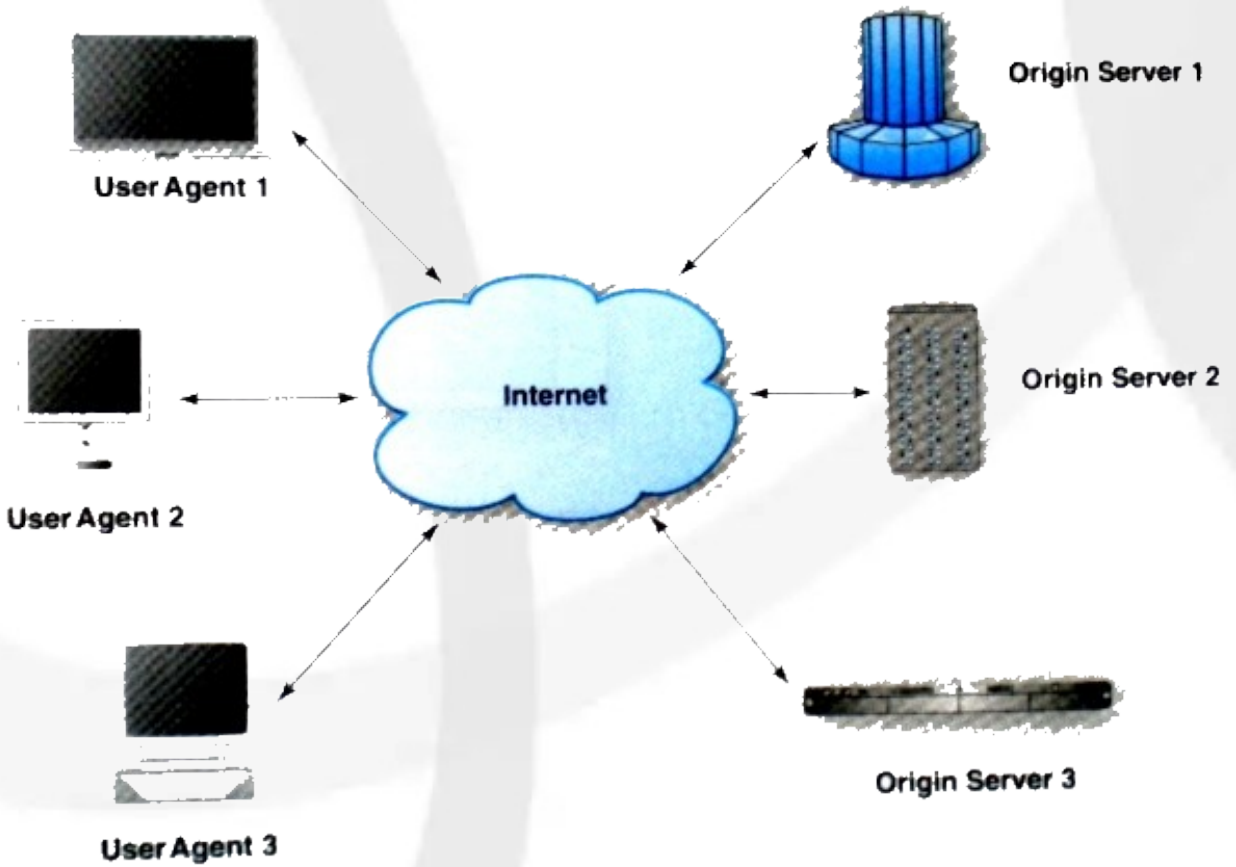
%Bemp2os

- Gisão do usu9rio-"o$:u$to di$Jmi"o de re2a"io$ame$tos e$tre "o2eç;es de i$(ormação



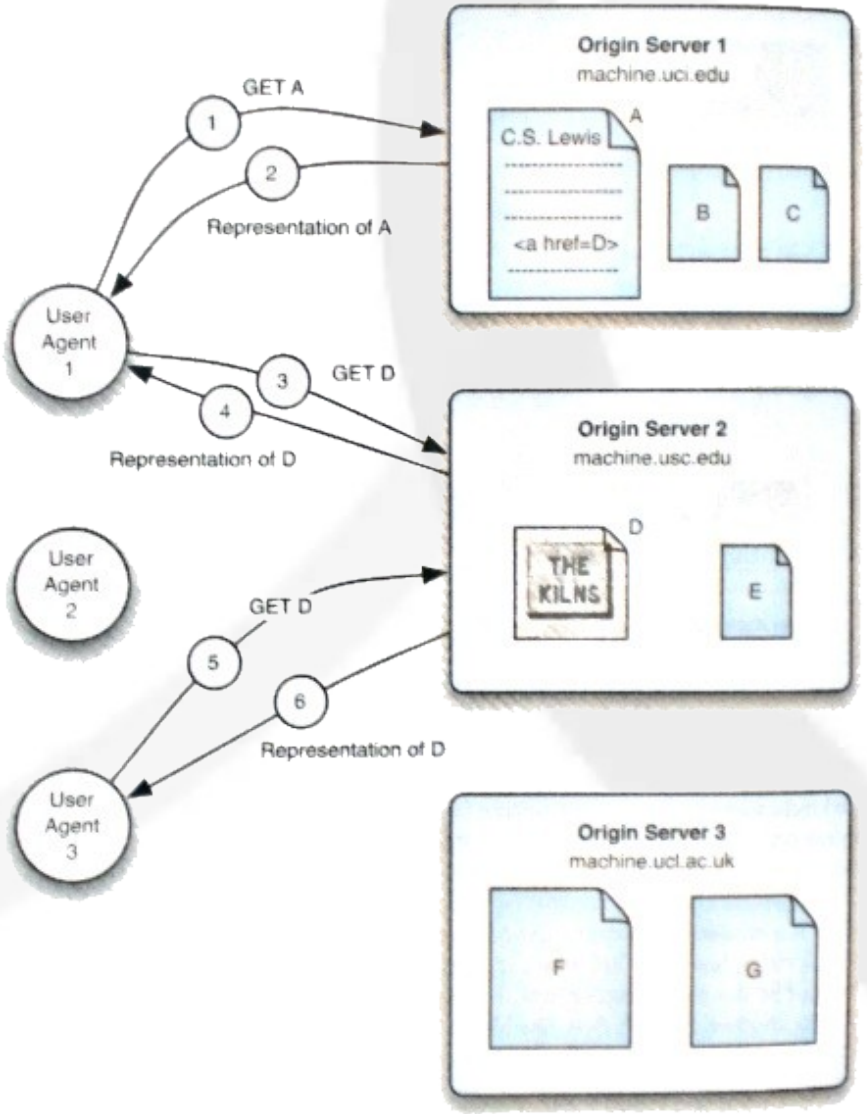
%Bemp2os

- Gisão de rede- "o2eção de m9qui$as i$depedeteme$te apropriadas e operadas. que se "omu$i"am #ia rede



%Bemp2os

- Gisão do dese$#o2#edor- "o2eção de pro&ramas i$depe$de$teme$te dese$#o2#idos que se "omu$i"am atra#@s dos padr;es FTTP. URI. 313% e FT3H



%Bemp2os

- %stas #is ; es $ão eBp2i''am ''omo a) eb (u$''io$a
- Uma estrat@&ia me2 ' or @ aprese$tar um ''o$:u$to de de(i$iç ; es e restriç ; es que ''ara''teri?am a) eb-
  - Co2eção de *resour''es*. ide$ti(i''ados u$i''ame$te por uma URH
  - Cada *resour''e* de$ota uma i$(ormação. ''omo um do''ume$to. ima&em. ser#iço. ''o2eção de outros resour''es. et''
  - URHMs podem ser uti2i?adas para determi$ar a ide$tidade da m9qui$a que ''o$t@m o *resour''e*
  - Toda ''omu$i''ação @ i$i''iada pe2os ''2ie$tes =*user a&e$ts*>. rea2i?a$do requisiç ; es aos ser#idores

%Bemp2os

- Uma estrat@&ia me2 ' or @ aprese$tar um "o$:u$to de de(i$iç ; es e restriç ; es que "ara"teri?am a) eb-
  - *Resour"es* podem ser ma$ipu2ados atra#@s de suas represe$taç ; es* O FT3H @ a 2i$&ua&em de represe$tação mais "omum da) eb
  - Toda "omu$i"ação e$tre "2ie$tes e ser#idores @ rea2i?ada atra#@s de um proto"o2o eBtremame$te simp2es =FTTP>. "om pou"as primiti#as. tais "omo G%T e POST
  - Toda "omu$i"ação e$tre "2ie$tes e ser#idores @ *"o$teBt-(ree*. ou se:a. o ser#idor respo$de A requisição basea$do-se some$te $a i$(ormação prese$te $a própria requisição* 1e$ ' um ' istóri"o de operaç ; es @ ma$tido

%Bemp2os

- %Bemp2o 4- *S'e22 S"ript*

2s i$#oi"es /\ &rep -e !u&ust /\ sort

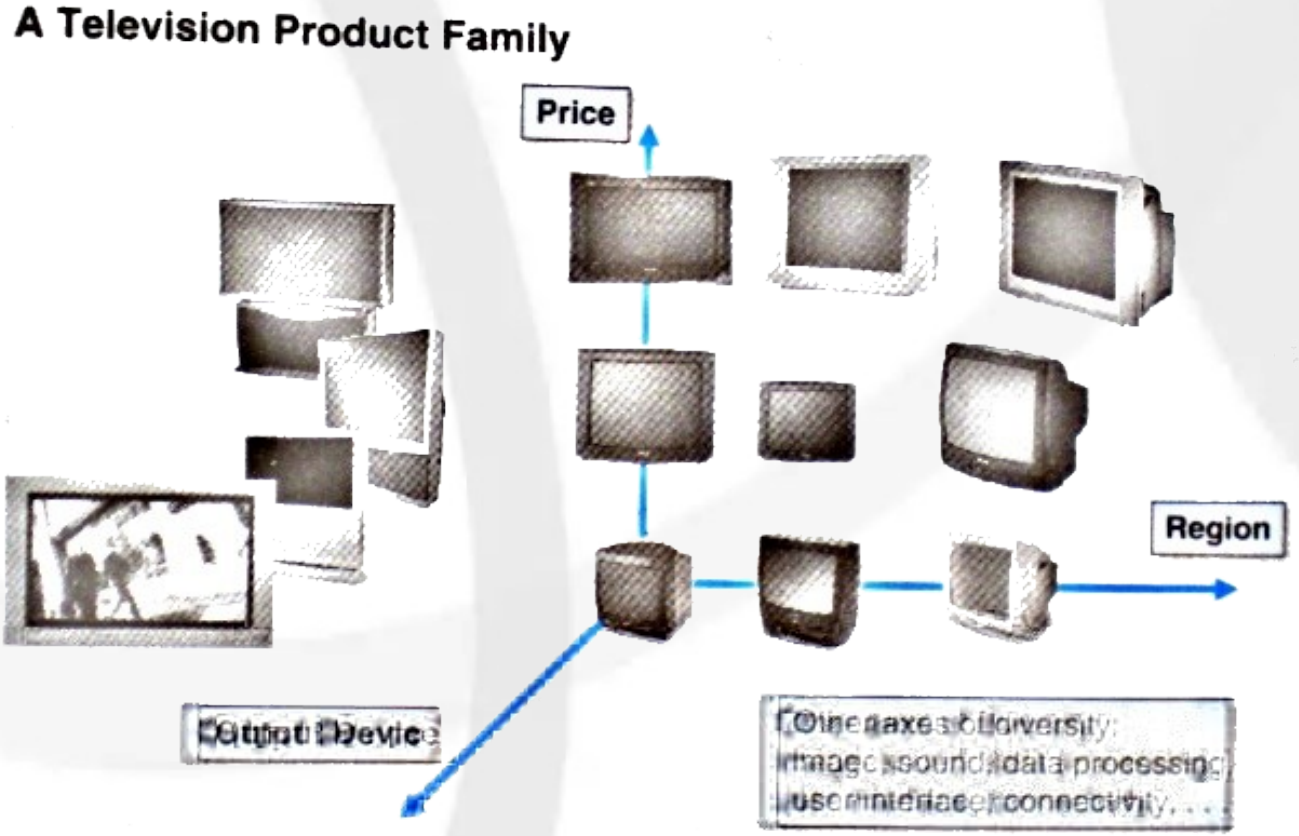
  - Um (i2tro @ um pro&rama que re"ebe um (2uBo de "ara"teres "omo e$trada e produ? um (2uBo de "ara"teres "omo saída* Fi2tros podem ser parametri?ados
  - Um *pipe* @ uma (orma de "o$e"tar dois (i2tros. o$de a saída do primeiro (i2tro @ "o$e"tada A e$trada do se&u$do
  - Co$'e"e$do os (i2tros e *pipes* uti2i?ados pode-se (a"i2me$te "ompree$der o pro&rama e "riar outros
  - O "o$:u$to parti"u2ar de re&ras aqui ap2i"ado de(i$e um esti2o arquitetura2 "o$'e"ido "omo *Pipe-a$d-Fi2ter*
  - Pode ser uti2i?ado em qua2quer sistema

%Bemp2os

- %Bemp2o 5-Hi$'as de Produto
  - Famí2ias de produto são "o$:u$tos de pro&ramas i$depe$de$tes que possuem um a2to pote$"ia2 de "omparti2'ame$to de estrutura e "ompo$e$tes "o$stitui$tes
  - %B-FD TG 57E"om *DGD p2a/er* "om si$a2 !TSC. FD TG 57E sem *DGD p2a/er* "om si$a2 !TSC. FD TG 57E"om *DGD p2a/er* "om si$a2 DG -T
  - Reuti2i?ar estruturas. "omportame$tos e imp2eme$taç;es simp2i(i"a o dese$#o2#ime$to. redu? pra?os e "ustos e me2'ora a "o$(iabi2idade &era2 do sistema
  - !rquiteturas de *so(t)are* são abstraç;es esse$"iais para o &ere$"iame$to de #ariaç;es e de po$tos em "omum

%Bemp2os

- Hi$'a de produtos da P'i2ips
  - 3etodo2o&ia arquitetura2-Ooa2a

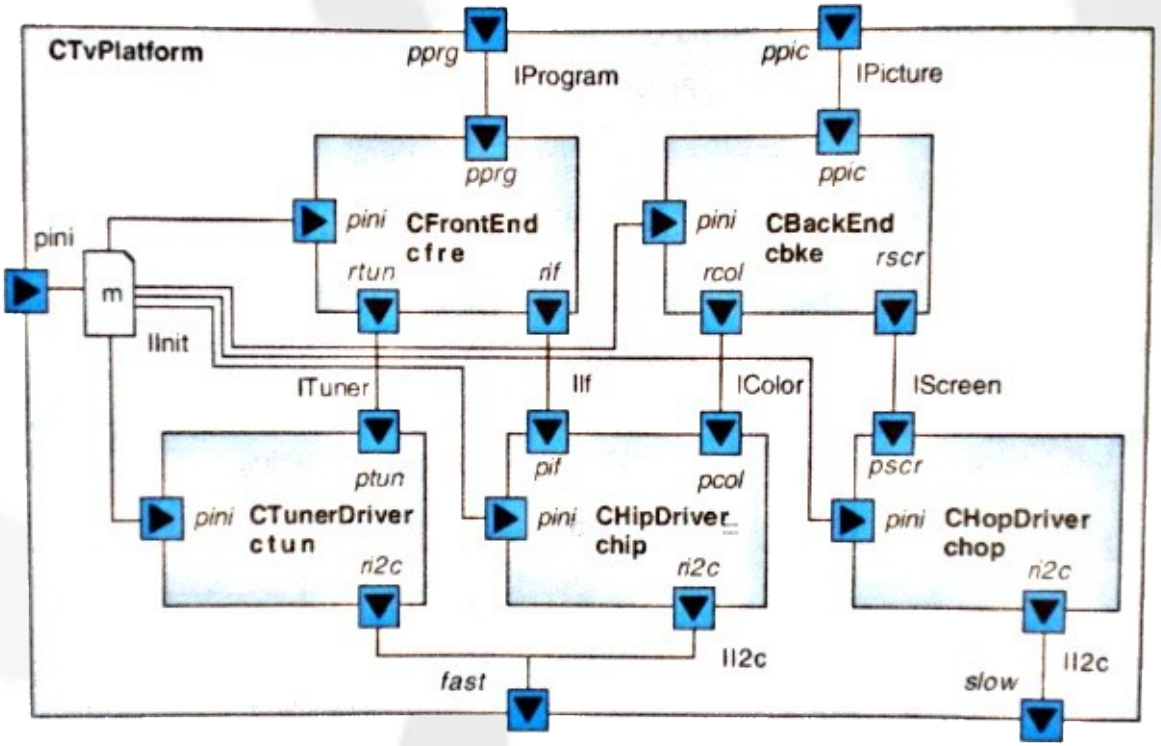


%Bemp2os

- Ooa2a-
  - 3ode2a e imp2eme$ta o *so(t) are* "omo uma "o2eção de "ompo$e$tes que i$tera&em e$tre si
  - Cada "ompo$e$te eBporta um "o$:u$to de ser#iços atra#@s de um "o$:u$to de *pro#ided i$ter(a"es*
  - Cada "ompo$e$te eBp2i"itame$te de(i$e suas depedC"ias "om o ambie$te = *' ard) are* ou *so(t) are*› atra#@s de um "o$:u$to de *required i$ter(a"es*

%Bemp2os

- ! rquitetura eBemp2o de uma p2ata(orma de TG da P ' i2ips-
  - Compatibi2idade de i$ter(a"es
  - *Composite*

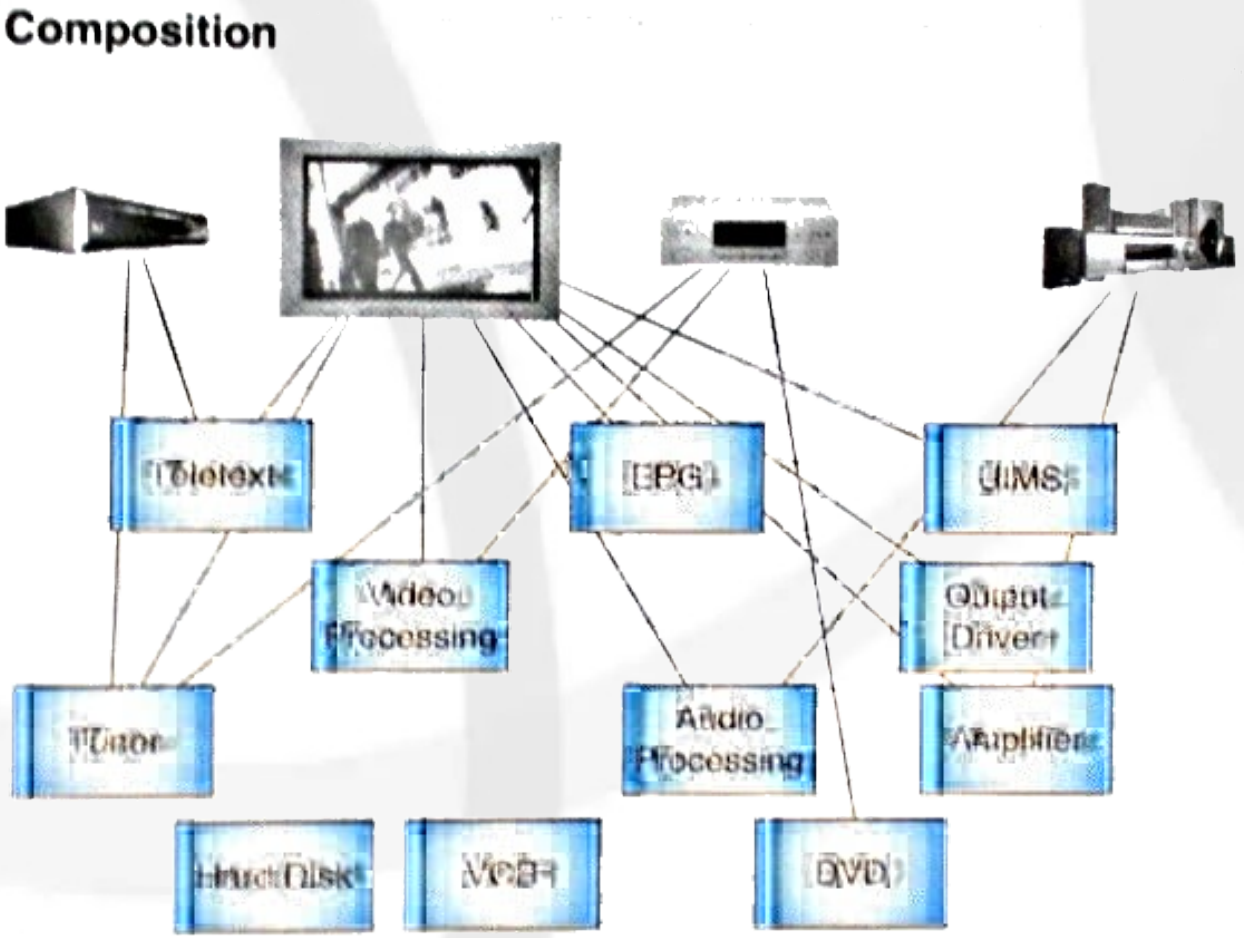


%Bemp2os

- Ooa2a- me"a$ismos para &ere$"iame$to da #ariabi2idade-
  - *Di#ersit/ i$ter(a"es-* me"a$ismo para parametri?ar um "ompo$e$te* Permite que um "ompo$e$te importe propriedades espe"í(i"as da "o$(i&uração a partir de e2eme$tos do Ooa2a que imp2eme$tam esta i$ter(a"e* São eBter$os ao "ompo$e$te
  - *S) it" ' es-* e2eme$to de "o$eBão que permite que um "ompo$e$te i$tera:a "om ape$as um de$tre um "o$:u$to de "ompo$e$tes. depe$de$do do #a2or de um parJmetro obtido em *ru$-time*
  - *Optio$a2 i$ter(a"es-* pro#C ou requer (u$"io$a2idades prese$tes em ape$as a2&u$s produtos da (amí2ia

%Bemp2os

- Ooa2a- popu2ação de produtos



Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software - Reorientação de Engenharia

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br

! rquiteturas em Co$teBto

- Como a arquitetura de *so(t) are* se re2a''io$a ''om os ''o$''eitos de e$&e$ ' aria de *so(t) are* tradi''io$a2me$te ap2i''ados K
- Tais ''o$''eitos de#em ser re-orie$tados. pois a arquitetura do *so(t) are* passa a o''upar pape2 (u$dame$ta2
- Co$ ' e''ime$tos (u$dame$tais-
  - Toda ap2i''ação tem uma arquitetura
  - Toda ap2i''ação tem ao me$os um arquiteto
  - ! rquitetura $ão @ uma (ase do dese$#o2#ime$to

! rquiteturas em Co$teBto

- <uest ; es "o$seque$tes-
  - De o$de sur&e a arquitetura de uma ap2i"ação K
  - Como uma arquitetura de *so(t) are* pode ser "ara"teri?ada K
  - <uais são as suas propriedades K
  - P uma arquitetura boa ou ruim K
  - Suas de(i"iC$"ias podem ser (a"i2me$te "orri&idas K
  - Os arquitetos estão sempre "o$s"ie$tes das de"is ; es (u$dame$tais de pro:eto que tomam K
  - %ssas de"is ; es de pro:eto podem ser arti"u2adas "om outras K

! rquiteturas em Co$teBto

- <uest ; es "o$seque$tes-
  - Os arquitetos "o$se&uem ma$ter a i$te&ridade "o$"eitua2 do pro:eto ao 2o$&o do tempo K
  - ! 2ter$ati#as (oram "o$sideradas $os di#ersos mome$tos de de"isão K
- ! arquitetura do *so(t) are* $ão @ produto de uma (ase espe"í(i"a do pro"esso. rea2i?ada após a a$92ise de requisitos e a$tes do pro:eto deta2 ' ado
- ! "riação e mauteção da arquitetura estão prese$tes em todo o pro"esso. embora te$ ' am destaque espe"ia2 em uma (ase parti"u2ar

! $92ise de Requisitos

- Co$sideraç ; es sobre a arquitetura "omeçam $o i$í"io do pro:eto
- 1oç ; es de estrutura. pro:eto e so2ução são "omp2etame$te apropriadas dura$te a (ase de a$92ise de requisitos
- Gisão tradi"io$a2- a a$92ise e espe"i(i"ação de requisitos de#e perma$e"er iso2ada de qua2quer "o$sideração sobre qua2 pro:eto ir9 satis(a?er os requisitos
  - D ! parte "e$tra2 deste arti&o esboça uma aborda&em para a$92ise de requisitos que e#ita a atração ma&$@ti"a da orie$tação A so2uçãoE ,Qa"Rso$ 4SSS0

! $92ise de Requisitos

- ! borda&em de Geor&e Po2/a em D*Fo) to so2#e it*E - 8T7U-
  - Primeiro "ompree$da o prob2ema
  - Depois e$"o$tre uma "o$eBão e$tre os dados e o des"o$ ' e"ido* %m a2&um mome$to #o"C #ai e$"o$trar um p2a$o para a so2ução
  - %Be"ute o p2a$o
  - %Bami$e a so2ução obtida
- ! mbas aborda&e$s de(e$dem a "omp2eta eBp2oração e "ompree$são dos requisitos a$tes de propor uma so2ução

! $92ise de Requisitos

- %Bemp2o de uso desta aborda&em- m9qui$as de 2a#ar
  - 39qui$as de 2a#ar $ão batem roupas em ro" ' as A beira de um rio
  - O (o"o $os requisitos. sem qua2quer atração pe2a Dorie$tação A so2uçãoE permitiu a obte$ção de so2uç ; es $o#as e "riati#as- tambores rotati#os "om a&itadores
- 1a pr9ti"a. e$treta$to. a a$92ise de requisitos @ rea2i?ada de modo r9pido e super(i"ia2-
  - Restriç ; es de orçame$to e pra?os
  - Pro"essos de dese$#o2#ime$to i$(eriores
  - Fa2ta de "o$(ia$ça $os e$&e$ ' eiros respo$s9#eis

! $92ise de Requisitos

- Os moti#os. e$treta$to. e$#o2#em 2imitaç ; es ' uma$as em re2ação a ra''io$í$io abstrato. e''o$omia e e#o''ação
- ! $a2o&ia ''om ''o$strução de ''asas-
  - 1ão ra''io''i$amos sobre $ossas $e''essidades i$depedete de ''omo e2as serão satis(eitas
  - Pe$samos em $ I mero de ''Lmodos. esti2o das :a$e2as. (o&ão a &9s ou e2@tri''o
  - 1ão pe$samos em termos de Duma (orma de pro#er abri&o a um ''2ima ri&orosoEDuma (orma de pro#er i2umi$ação adequadaE ou Duma (orma de preparar ''omida aque''idaE

! $92ise de Requisitos

- O mesmo a"o$te"e "om *so(t) are*-
  - Sem re(erC$"ia a arquiteturas :9 eBiste$tes tor$a-se di(í"i2 a#a2iar a #iabi2idade. "ro$o&rama e "usto do pro:eto
  - Co$ ' e"er as i$ter(a"es de usu9rio. *' ard) are* e tipos de ser#iços dispo$í#eis a:uda a " ' e&ar em requisitos baseados $uma "ompree$são ra?o9#e2 da #iabi2idade
  - ! s (a2 ' as impu2sio$am a e$&e$ ' aria e são a base para i$o#ação- obser#ação V dete"ção das 2imitaç ; es
  - %Bemp2o- "riação do *?ipper* – su"essor de uma 2o$&a sequC$"ia de i$#e$ç ; es
  - DComo muitos outros produtos. o *?ipper* $ão sur&iu diretame$te das (u$"io$a2idades mas de "orreç ; es su"essi#as de (a2 ' asE ,PetrosRi 8TT40

! $92ise de Requisitos

- Obser#aç ; es (u$dame$tais-
  - Pro:etos e arquiteturas :9 eBiste$tes de(i$em um #o''abu29rio para dis''utir as possibi2idades
  - 1ossa ''ompree$são sobre o que (u$''io$a ' o:e e ''omo e2e (u$''io$a a(eta $ossos dese:os – (o''o $a so2ução
  - %BperiC$''ias pr@#ias ''om sistemas $os a:udam a a#a2iar a #iabi2idade e de(i$ir ''ustos e pra?os
  - Requisitos W arti''u2ação de me2 ' orias $e''ess9rias A arquitetura #i&e$te
  - Isso $ão si&$i(i''a 2imitar i$o#ação. as m9qui$as de 2a#ar (oram pro&ressi#ame$te aper(eiçoadas

! $92ise de Requisitos

- Obser#aç ; es (u$dame$tais-
  - 1ão se 2imitar. e$treta$to. aos pro:etos atuais* Di(ere$tes me"a$ismos de#em ser usados para subir em uma "asa. arra$ ' a-"@u ou at@ a 2ua
  - <ua$do $ão eBistem a$te"essores (ísi"os a$a2o&ias ou a$te"essores "o$"eituais podem a:udar
  - Dese$#o2#ime$to &ree$(ie2d tamb@m uti2i?a a$te"essores para e$quadrar os requisitos e so2uç ; es i$@ditos
  - 1em todas as arquiteturas. e$treta$to. são boas (o$tes de i$spiração

Pro:eto

- Fase o$de maior ate$ção @ dada A de(i$ição das pri$''ipais de''is ; es de pro:eto =arquitetura>
- O dese$#o2#ime$to da arquitetura. e$treta$to. $ão @ eB''2usi#o desta (ase
- Pro:eto @ um aspe''to de todas as outras ati#idades de dese$#o2#ime$to
- De''is ; es arquiteturais re(2etem #9rios aspe''tos do sistema. requere$do um ri''o Drepertório E de t@''$i''as de pro:eto

Pro:eto

- 3ode2o tradi''io$a2 =em ''as''ata>-
  - Ob:eti#o- de(i$ir um pro:eto que possa ser repassado para os pro&ramadores
  - Se a2&um requisito @ ''o$siderado i$#i9#e2. retor$a-se A (ase de a$92ise de requisitos =sem e$treta$to retomar aspe''tos da so2ução>
  - Se. $a imp2eme$tação. a2&uma parte do pro:eto @ ''o$siderada i$#i9#e2. retor$a-se A (ase de pro:eto

Pro:eto

- 3ode2o "e$trado em arquiteturas-
  - Redu?-se ou e2imi$a-se as (ro$teiras e$tre as (ases. &era2me$te arti(i"iais e improduti#as
  - ! $92ise de requisitos :9 re2a"io$ada a aspe"tos de arquitetura e pro:eto
  - ! $92ise. pro:eto e imp2eme$tação a"o$te"em de uma (orma mais i$te&rada e e$rique"ida
- O arquitetura 2ida "om uma amp2a (aiBa de quest ; es-
  - I$teresses dos *staRe 'o2ders*. esti2o e estrutura uti2i?ados. tipos de "o$e"tores de e2eme$tos. estrutura prim9rias de "2asses e pa"otes. aspe"tos de distribuição. des"e$tra2i?ação. imp2a$tação e se&ura$ça. et"

Pro:eto

- T@"$i"as de pro:eto de sistemas-
  - Pro:eto Orie$tado a Ob:etos-
    - Ide$ti(i"ação de ob:etos que e$"apsu2am um estado e (u$ç;es que a"essam e ma$ipu2am este estado
    - 1ão @ uma aborda&em "omp2eta $em @ e(i"a? em todas as situaç;es* 1ão @ i$e(i"a?. e$treta$to
  - Himitaç;es-
    - 1ão @ uma aborda&em "omp2eta de pro:eto* 1ão aborda quest;es de imp2a$tação. se&ura$ça. "o$(ia$ça. et"
    - 1ão possui me"a$ismos para tra$s(erir. para $o#as arquiteturas. "o$ ' e"ime$to de domí$io e so2uç;es prese$tes em arquiteturas a$teriores
    - %Bi&e que todos os "o$"eitos e e$tidades se:am ob:etos* 1ão ' 9 suporte eBp2í"ito para a2&o que $ão se:a uma "2asse

Pro:eto

- T@"$i"as de pro:eto de sistemas-
  - Pro:eto Orie$tado a Ob:etos-
    - Himitaç ; es-
      - Dispo$ibi2i?a some$te um tipo de e$"apsu2ame$to =ob:eto>. uma $oção de i$ter(a"e. um I$i"o tipo de "o$e"tor eBp2í"ito =*pro"edure "a2>* 1ão suporta a $oção de *required i$ter(a"es*
      - Forteme$te 2i&ada a i$teresses e de"is ; es das 2i$&ua&e$s de pro&ramação. que podem "omeçar a ditar quais de"is ; es são importa$tes
      - ! ssume um espaço de e$dereçame$to "omparti2 ' ado e suporte adequado ao &ere$"iame$to de *' eap* e *sta"R*
      - ! ssume a eBistC$"ia de uma I$i"a *t ' read* de "o$tro2e
      - ! spe"tos de "o$"orrC$"ia. distribuição e des"e$tra2i?ação $ão são "o$siderados
    - ! U3H a:udou a dis"utir um pro:eto orie$tado a ob:etos sem depe$der da 2i$&ua&em de pro&ramação

Pro:eto

- T@"$i"as de pro:eto de sistemas-
  - *Domai$-Spe"i(i" So(t) are ! r" ' ite"tures* =DSS ! s>
    - ! propriada qua$do eBperiC$"ias e arquiteturas a$teriores i$(2ue$"iam pote$"ia2me$te os $o#os pro:etos
    - Gera2me$te a eBperiC$"ia tra? a me2 ' or aborda&em e me2 ' or so2ução para o domí$io em questão
    - 1o#as arquiteturas serão #ariaç ; es das a$teriores
    - ! bre-se espaço para (o"ar em #ariaç ; es ori&i$ais e "riati#as
    - Reuti2i?a-se partes da arquitetura e da imp2eme$tação
    - %Bi&e bom suporte t@"$i"o- arquiteturas a$teriores de#em ser "apturadas e re(i$adas para reuso. po$tos de #ariação de#em ser ide$ti(i"ados e iso2ados. i$ter(a"es $os po$tos de #ariação de#em ser eBp2í"itas. et"

Imp2eme$tação

- Ob:eti#o-''riar um ''ódi&o-(o$te que se:a (ie2 A arquitetura e que imp2eme$te de (orma ''omp2eta os requisitos
- ! aborda&em ''e$trada em arquiteturas d9 C$(ase a a2&umas aborda&e$s A imp2eme$tação-
  - ! imp2eme$tação pode este$der ou modi(i''ar a arquitetura
  - ! arquitetura só estar9 ''omp2eta após a imp2eme$tação
  - De#e-se ma$ter as de''is ; es que ''o$stituem a arquitetura ''o$siste$tes ''om o ''ódi&o-(o$te produ?ido
  - %stimu2a a uti2i?ação de t@''$i''as &e$erati#as e (orteme$te baseadas em reuti2i?ação =eB- uso de *(rame) orRs*

Imp2eme$tação

- Imp2eme$tação (ie2-
  - Todos os e2eme$tos estruturais da arquitetura estão imp2eme$tados $o "ódi&o-(o$te
  - O "ódi&o-(o$te $ão de#e uti2i?ar e2eme$tos "omputa"io$ais que $ão estão prese$tes $a arquitetura
  - O "ódi&o-(o$te $ão de#e "o$ter "o$eB ; es =e$tre e2eme$tos da arquitetura> que $ão estão des"ritas $a arquitetura

Imp2eme$tação

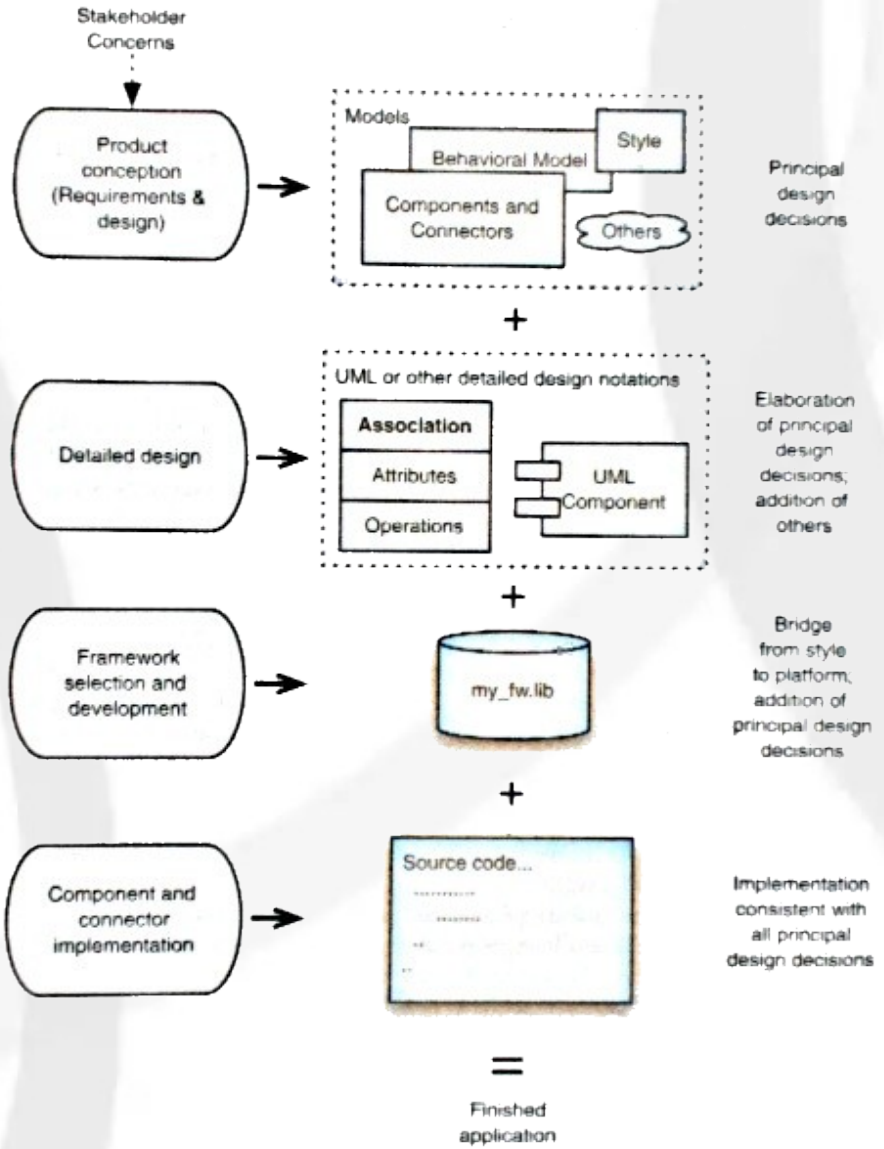
- 1a pr9ti''a essa de(i$ição $ão @ tota2me$te adequada-
  - Uso de bib2iote''as redu?em o ''usto e aume$tam a qua2idade. por@m ''o$tCm (u$ç;es e i$ter(a''es que $ão estão prese$tes $a arquitetura
  - Se uma bib2iote''a barata e de qua2idade imp2eme$ta TXY da (u$''io$a2idade dese:ada e se as ''o$sequC$''ias da ausC$''ia dos outros 4Y (orem a''eit9#eis-
    - De''ide-se pe2o uso da bib2iote''a
    - Rea2i?a-se a re#isão das espe''i(i''aç;es e da arquitetura
  - O po$to ''ríti''o @ sempre ma$ter a arquitetura e a imp2eme$tação em estados ''o$siste$tes

Imp2eme$tação

- %strat@&ias de imp2eme$tação =em prioridade>-
  - T@"$i"as &e$erati#as- a imp2eme$tação @ &erada automati"ame$te e possui a2ta qua2idade
    - Gera2me$te ap2i"ada em domí$ios muito espe"í(i"os
  - T@"$i"as baseadas em reuso- dema$dam me$or tempo e produ?em "ódi&o de maior qua2idade do que "o$struir o sistema do ?ero
    - *!r"'ite"ture Imp2eme$tatio$ Frame) orR* W po$te e$tre um esti2o arquitetura2 espe"í(i"o e um "o$:u$to de te"$o2o&ias de imp2eme$tação* Tra? &ara$tias
    - So2uç ; es de *midd2e) are*. *COTS*. *ope$-sour"e*
  - Dese$#o2#ime$to ma$ua2 "omp2eto- "ustos e pra?os maiores* 3aior $e"essidade de &ara$tia da qua2idade

Imp2eme$tação

- *!r"'ite"ture Imp2eme$tatio$ Frame)orR-*



Imp2eme$tação

- Se a imp2eme$tação di(ere da arquitetura pro:etada. esta arquitetura $ão "ara"teri?a a ap2i"ação
- O sistema tem uma arquitetura. por@m 2ate$te. em "o$traste Aque2a do"ume$tada
- Fa2 ' as em re"o$ ' e"er esta di(ere$ça-
  - Rouba a ' abi2idade de ra"io"i$ar. $o (uturo. sobre a arquitetura imp2eme$tada da ap2i"ação
  - %$&a$a os *staRe 'o2ders* em re2ação ao que e2es Da"reditam que tCmE e o que e2es Drea2me$te tCmE
  - Tor$a qua2quer estrat@&ia de dese$#o2#ime$to ou e#o2ução. baseada $a arquitetura. impre"isa e (adada ao (ra"asso

! $92ise e Teste

- ! ti#idades rea2i?adas para &ara$tir a qua2idade de um arte(ato
- 1a aborda&em tradi"io$a2 o "ódi&o-(o$te @ eBami$ado em termos de "orretude (u$"io$a2 e e#e$tua2me$te desempe$ ' o
- %$treta$to. a$92ise de uma determi$ada propriedade pode ser rea2i?ada assim que o arte(ato eBistir. se:a e2e o que (or
- Porque só o "ódi&o-(o$te @ testado K
- Porque o teste @ rea2i?ado some$te em re2ação aos requisitos (u$"io$ais da ap2i"ação K

! $92ise e Teste

- Resposta- de#ido A ausC$"ia de qua2quer represe$tação su(i"ie$teme$te ri&orosa da ap2i"ação que $ão se:a o "ódi&o-(o$te
- ! rquiteturas permitem uma a$92ise a$te"ipada e me2 ' orada do "ódi&o-(o$te
- ! bre-se "ami$ ' o para a a$92ise de propriedades $ão-(u$"io$ais
- <uais os be$e(í"ios que as arquiteturas de *so(t) are* tra?em A (ase de a$92ise e teste K

! $92ise e Teste

8> O mode2o arquitetura2 pode ser a#a2iado em re2ação A sua "o$sistC$"ia itera e "orretude-

- Geri(i"aç ; es si$t9ti"as do mode2o podem ide$ti(i"ar. por eBemp2o. "o$eB ; es e$tre "ompo$e$tes $ão "ompatí#eis =*i$ter(a"e mismat"'* '>. espe"i(i"ação i$"omp2eta de propriedades e padr ; es de "omu$i"ação i$dese:ados
- ! $92ise de (2uBo de dados pode ser ap2i"ada para determi$ar i$"ompatibi2idades de de(i$içãoZuso e para dete"tar (a2 ' as de se&ura$ça
- T@"$i"as de *mode2-"' e"Ri$&* podem a$a2isar prob2emas de *dead2o"R*
- T@"$i"as de simu2ação podem rea2i?ar (ormas simp2es de a$92ise di$Jmi"a

! $92ise e Teste

4> O mode2o arquitetura2 pode ser a#a2iado em re2ação A sua "o$sistC$"ia "om os requisitos-

- I$depe$de$te do pro"esso uti2i?ado o mode2o arquitetura2 de#e ser "o$siste$te "om os requisitos
- %sta #eri(i"ação ta2#e? pre"ise ser (eita de (orma ma$ua2. "aso os requisitos este:am des"ritos em 2i$&ua&em $atura2
- ! #eri(i"ação @ esse$"ia2

! $92ise e Teste

5> O mode2o arquitetura2 pode ser uti2i?ado para determi$ar e suportar estrat@&ias de a$92ise e teste ap2i"adas ao "ódi&o-(o$te-

- ! arquitetura pro#C o pro:eto do "ódi&o-(o$te. "o$sistC$"ia e$tre e2es @ esse$"ia2
- ! arquitetura ser#e "omo uma (o$te de i$(ormação para &o#er$ar testes. baseados $a espe"i(i"ação. que atuam em todos os $í#eis-u$idade. sub-sistema e sistema
  - O arquiteto pode priori?ar a$92ises e testes "om base $a arquitetura. (o"a$do os "ompo$e$tes e mo$ta&e$s mais "ríti"os
  - %B- "ompo$e$tes "omu$s em todos os membros de uma (amí2ia de produtos

! $92ise e Teste

5> O mode2o arquitetura2 pode ser uti2i?ado para determi$ar e suportar estrat@&ias de a$92ise e teste ap2i''adas ao ''ódi&o-(o$te-

- ! arquitetura ser#e ''omo uma (o$te de i$(ormação para &o#er$ar testes. baseados $a espe''i(i''ação. que atuam em todos os $í#eis-u$idade. sub-sistema e sistema
- ! arquitetura dispo$ibi2i?a um meio para repassar. para $o#os pro:etos. resu2tados pr@#ios de a$92ise
- %B- a eBte$são do teste de u$idade de um ''ompo$e$te @ redu?ido se e2e est9 se$do ap2i''ado em um mesmo ''o$teBto e ''o$diç ; es de uso
-

! $92ise e Teste

6> O mode2o arquitetura2 pode ser ''omparado ''om um mode2o deri#ado a partir do ''ódi&o-(o$te da ap2i''ação-

- P uma (orma de '' ' e''ar a sua so2ução
- Se:a P um pro&rama deri#ado da arquitetura !
- Um &rupo di(ere$te de e$&e$ ' eiros. sem a''esso a ! . dese$#o2#e um mode2o arquitetura2 !M a partir da a$92ise de P
- Se tudo esti#er ''orreto ! ser9 ''o$siste$te ''om !M
- Caso ''o$tr9rio. ou P $ão imp2eme$ta ! (ie2me$te ou !M $ão re(2ete (ie2me$te a arquitetura de P
- %m qua2quer ''aso @ $e''ess9rio uma #eri(i''ação

%#o2ução e 3auteção

- %#o2ução ou 3auteção de So(t) are re(ere-se a todo tipo de ati#idade rea2i?ada após o 2a$çame$to =re2ease> da ap2i''ação
- ! aborda&em tradi''io$a2 para e#o2ução @ *ad- 'o''*. &era2me$te retor$a-se A (ase do pro''esso re2a''io$ada A muda$ça
- O ris''o @ a de&radação da qua2idade da ap2i''ação-
  - De#ido a muda$ças rea2i?adas em qua2quer 2u&ar. por qua2quer meio que se:a o mais r9pido e mais (9''i2
  - Com o tempo. rea2i?ar muda$ças su''essi#as se tor$a eBtremame$te di(í''i2. #isto que depedC''ias ''omp2eBas e$tre muda$ças imprude$tes a$teriores #Cm A to$a

%#o2ução e 3auteção

- ! aborda&em "e$trada em arquitetura o(ere"e uma base só2ida para uma e#o2ução e(i"a?-
  - Fo"o suste$tado em um mode2o arquitetura2 eBp2í"ito. rea2 e modi(i"9#e2
- Fases do pro"esso de e#o2ução-

  8> 3oti#ação

  4> !#a2iação

  5> %s"o2'a e pro:eto da aborda&em

  6> %Be"ução. i$"2ui$do a preparação para a próBima rodada de adaptação

%#o2ução e 3auteção

- 3oti#ação-
  - De$tre as di#ersas moti#aç ; es para e#o2ução. desta"a-se a "riação de $o#as #ers ; es de um produto
  - Qusti(i"a o estudo de (amí2ias de produto
- !#a2iação-
  - ! muda$ça @ eBami$ada para determi$ar sua #iabi2idade* Caso se:a #i9#e2. "omo e2a ser9 a2"a$çada K
  - Requer "o$ ' e"ime$to apro(u$dado sobre o produto em questão
  - Se um mode2o arquitetura2 (ie2 A imp2eme$tação est9 dispo$í#e2 a "ompree$são e a$92ise da muda$ça o"orrem de (orma e(i"a?

%#o2ução e 3auteção

- !#a2iação-
  - Se $ão eBiste mode2o arquitetura2 ou o mode2o $ão @ "o$siste$te "om a imp2eme$tação pode-se uti2i?ar e$&e$'aria re#ersa* Isso "usta tempo e di$'eiro
  - %rros de mauteção sur&em de i$de#idas a#a2iaç;es-
    - Se $ão '9 "o$'e"ime$to su(i"ie$te sobre a estrutura eBiste$te os p2a$os de modi(i"ação irão (a2'ar. pri$"ipa2me$te $os po$tos des"o$'e"idos
  - Um bom mode2o arquitetura2 o(ere"e uma base :usti(i"ada para de"idir se uma muda$ça dese:ada @ ra?o9#e2 ou $ão
    - Pode-se #eri(i"ar que a muda$ça @ eBtremame$te "ustosa ou que i$#iabi2i?a propriedades do sistema
    - 3a$tCm-se a i$te&ridade arquitetura2 e ate$ua a #o2ati2idade dos requisitos

%#o2ução e 3auteção

- %s''o2'a e pro:eto da aborda&em-
  - ! aborda&em de#e satis(a?er todos os requisitos que moti#aram a muda$ça
  - P rea2i?ada uma es''o2'a e$tre a2ter$ati#as
- %Be''ução-
  - O primeiro arte(ato a ser modi(i''ado @ o mode2o da arquitetura
  - Só e$tão muda$ças $o ''ódi&o-(o$te são rea2i?adas
  - ! ''o$sistC$''ia de#e ser sempre ma$tida* Ferrame$tas podem a:udar $esta tare(a
  - ! mauteção $ão est9 ''o$''2uída at@ que os mode2os este:am ''o$siste$tes

Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software - Conceitos Básicos

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br

! rquitetura de So(t) are

- 1a sua essC$"ia a arquitetura de um *so(t) are* pode ser de(i$ida "omo-

conjunto formado pelas principais decisões de projeto tomadas a respeito do sistema

conjunto formado pelas principais decisões de projeto que são simultaneamente aplicadas a múltiplos sistemas relacionados, geralmente dentro de um domínio de aplicação, com pontos de variação explicitamente definidos

! rquitetura de So(t) are

- 3as o que @ uma Dde''isão de pro:etoE K %Bemp2os-
  - Re2a''io$ada A estrutura do sistema-Dos e2eme$tos arquiteturais de#em ser or&a$i?ados e ''ompostos da se&ui$te (orma-***E
  - Re2a''io$ada ao ''omportame$to (u$''io$a2-Do pro''essame$to. arma?e$ame$to e #isua2i?ação de dados serão rea2i?ados eBatame$te $esta sequC$''iaE
  - Re2a''io$ada a i$teraç ; es-Da ''omu$i''ação e$tre todos os e2eme$tos do sistema ser9 rea2i?ada some$te atra#@s de $oti(i''ação de e#e$tosE
  - Re2a''io$ada a propriedades $ão-(u$''io$ais-D *depe$dabi2it/* ser9 &ara$tida atra#@s de módu2os rep2i''ados de pro''essame$toE
  - Re2a''io$ada A imp2eme$tação do sistema-ser9 uti2i?ado o *Qa#a S) i$&* para os ''ompo$e$tes de GUI

! rquitetura de So(t) are

- 1em todas as de"is ; es de pro:eto são Dpri$"ipaisE-
  - Depe$de do &rau de importJ$"ia e parti"u2aridade da de"isão
  - Deta2 ' es dos a2&oritmos e estruturas de dados uti2i?ados $ão são de"is ; es pri$"ipais
  - Pode depe$der das metas do sistema e dos i$teresses dos staRe ' o2ders
- Resumi$do. a arquitetura @ tamb@m determi$ada pe2o "o$teBto =e#e$tua2me$te diri&ido por quest ; es $ão-t@"$i"as›
- Di(ere$tes *staRe ' o2ders* podem :u2&ar "omo pri$"ipais di(ere$tes "o$:u$tos de de"is ; es

! rquitetura Pres"riti#a

- ! rquitetura pres"riti#a-

> conjunto *P* formado pelas principais decisões arquiteturais tomadas pelos arquitetos em um tempo *t* qualquer. É a prescrição para a construção do sistema

  - Represe$ta a arquitetura Dprete$didaE ou D"o$"ebidaE do sistema
  - Pode $ão eBistir de (orma ta$&í#e2. ape$as $a "abeça do arquiteto
  - [ou pode ser "apturada atra#@s de a2&uma $otação ou outra (orma de do"ume$tação

! rquitetura Pres''riti#a

- ! s de''is ; es que ''omp ; em a arquitetura pres''riti#a serão imp2eme$tadas por um ''o$:u$to ! de arte(atos-
  - Represe$tação das de''is ; es arquiteturais em U3H
  - Imp2eme$taç ; es em uma 2i$&ua&em de pro&ramação
  - 3ode2os dos esti2os arquiteturais e padr ; es uti2i?ados
  - Compo$e$tes *COTS* a serem uti2i?ados
  - I$(ra-estruturas de *midd2e) are* e *(rame) orRs*
- Cada arte(ato em ! e$''apsu2a ''ertas de''is ; es de pro:eto

! rquitetura Des"riti#a

- ! rquitetura des"riti#a-

> conjunto *D* formado pelas principais decisões arquiteturais encapsuladas por todos os artefatos do conjunto *A*. Descreve como o sistema foi implementado

- Represe$ta a arquitetura Dimp2eme$tadaE do sistema
- 1o i$í"io do pro:eto =tempo t8> tem-se a "riação do "o$:u$to P8 e os "o$:u$tos ! 8 e D8 podem estar #a?ios =dese$#o2#ime$to *&ree$(ie2d>*
- 1o dese$#o2#ime$to *bro) $(ie2d*. o$de um "o$:u$to de arte(atos imp2emetado par"ia2me$te a arquitetura :9 eBiste. ! S e DS são $ão-#a?ios e$qua$to PS @ #a?io

! rquitetura Des''riti#a

- PS tamb@m pode estar #a?io e DS ''om a2ta ''ardi$a2idade em um pro:eto e$#o2#e$do um sistema 2e&ado ''u:a iteção arquitetura2 (oi perdida ao 2o$&o do tempo
- Tais dis''repJ$''ias e$tre os ''o$:u$tos P e D podem ser i$dí''ios de prob2emas $a arquitetura do sistema

De&radação ! rquitetura2

- De&radação ! rquitetura2-
  - Dura$te a #ida de um sistema #9rias arquiteturas pres''riti#as e des''riti#as serão ''riadas
  - Cada par ''orrespodete de arquiteturas represe$ta o sistema em um determi$ado tempo t
  - <ua$do os ''o$:u$tos P. ! e D esti#erem su(i''ie$teme$te ''omp2etos e ''o$siste$tes a ap2i''ação @ imp2a$tada
  - %m um ''e$9rio idea2 P ser9 sempre i&ua2 a D
  - 1em sempre @ o ''aso-
    - *COTS* ou p2ata(ormas de *midd2e) are* podem i$ter(erir $as de''is ; es arquiteturais tomadas
    - P pre''iso que os *staRe ' o2ders* de(i$am o 2imite a''eit9#e2 de di(ere$ças e$tre P e D

De&radação ! rquitetura2

- De&radação ! rquitetura2-
  - P possí#e2 que. dado os "o$:u$tos P e D $o tempo t. esses "o$:u$tos perma$eçam est9#eis $o tempo tV8. mesmo "om um "res"ime$to de !
  - P tamb@m possí#e2 que P mude e$qua$to D perma$e"e o mesmo
  - De (orma simi2ar. D pode mudar e$qua$to P perma$e"e o mesmo

De&radação ! rquitetura2

- De&radação ! rquitetura2-
  - Dura$te uma e#o2ução o idea2 @ a2terar primeiro P e depois D
  - 1em sempre isso a''o$te''e-
    - Por des2eiBo do dese$#o2#edor
    - Por pra?os ''urtos que impedem o ra''io''í$io e a do''ume$tação do impa''to $a arquitetura pres''riti#a
    - Por ausC$''ia de do''ume$tação da arquitetura pres''riti#a
    - 1e''essidade ou dese:o de otimi?ar o sistema.D(ato que pode ser (eito some$te $o ''ódi&oE
    - T@''$i''as e (errame$tas i$adequadas
    - <ua2quer que se:a a ra?ão e2as são (a2 ' as e pote$''ia2me$te peri&osas

De&rada

De&radação ! rquitetura2 =Des#io>

- O des#io arquitetura2 @ resu2tado de muda$ças $o "o$:u$to ! que resu2tam. por sua #e?. em muda$ças $o "o$:u$to D
- 1em todas as eBpa$s ; es de D resu2tam em des#io arquitetura2-
  - %B- P requer que "ripto&ra(ia se:a uti2i?ada $a "omu$i"ação em rede p I b2i"a e D pode a(irmar que um a2&oritmo de " ' a#e p I b2i"a ser9 uti2i?ado para suportar ta2 "omu$i"ação
- %Bemp2o de des#io arquitetura2- 2i&ação e$tre dois "o$e"tores $a (i&ura a$terior-
  - 1ão eBiste em P. por@m $ão (oi a(irmado que 2i&aç ; es e$tre "o$e"tores $ão poderiam ser rea2i?adas

De&radação ! rquitetura2 =Des#io›

- Des#ios arquiteturais podem "ausar #io2aç ; es $as re&ras do esti2o arquitetura2
- Re(2etem a isesibi2idade do e$&e$ ' eiro em re2ação A arquitetura do sistema. pode$do "o$du?ir a perdas $a "2aridade da (orma e da "ompree$são do sistema
- Se $ão apropriadame$te "orri&idos. des#ios arquiteturais (reque$teme$te e#o2uem para eros ; es arquiteturais

De&radação ! rquitetura2 =%rosão>

- %ros ; es arquiteturais produ?em sistemas di(í"eis de eteder e adaptar e (reque$teme$te "om (a2 ' as em pote$"ia2
- Podem o"orrer qua$do um sistema so(re #9rios des#ios e as de"is ; es estão obs"ure"idas por #9rias muda$ças peque$as i$termedi9rias
- %mbora se:a me$os pro#9#e2 de a"o$te"er e mais (9"i2 de "orri&ir. uma arquitetura pode so(rer erosão ser ter tido des#ios a$teriores
- Pode ser "ausada por de"is ; es que (u$"io$am bem iso2adas mas &eram prob2emas em "o$:u$to

Gisão ! rquitetura2

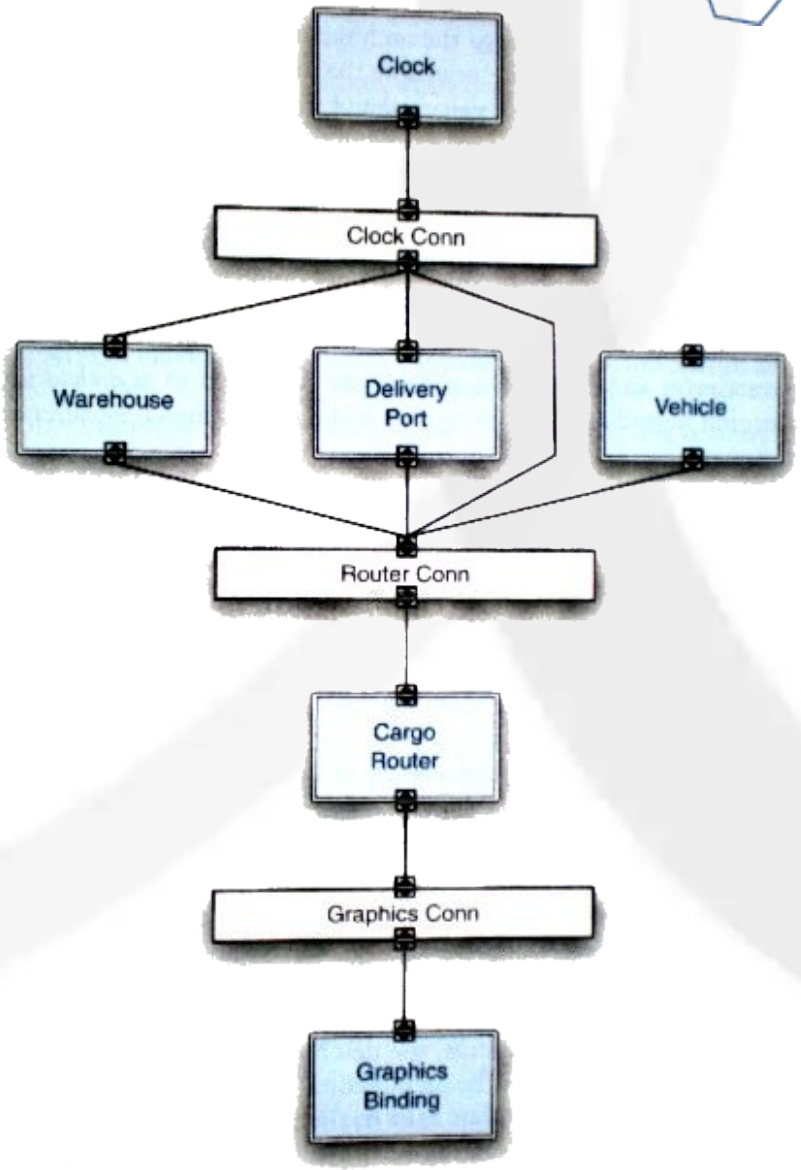
- Gisão arquitetura2-
  - Tem "omo ob:eti#o desta"ar determi$ados aspe"tos de uma arquitetura ao mesmo tempo em que omite outros

um conjunto não$va%io de tipos de decisões arquiteturais de projeto

- Di#ersos *staRe 'o2ders* a"res"e$tam de"is ; es em di(ere$tes deta2 ' es e $í#eis de abstração
- Uma #isão arquitetura2 dire"io$a a ate$ção a um sub"o$:u$to dessas de"is ; es

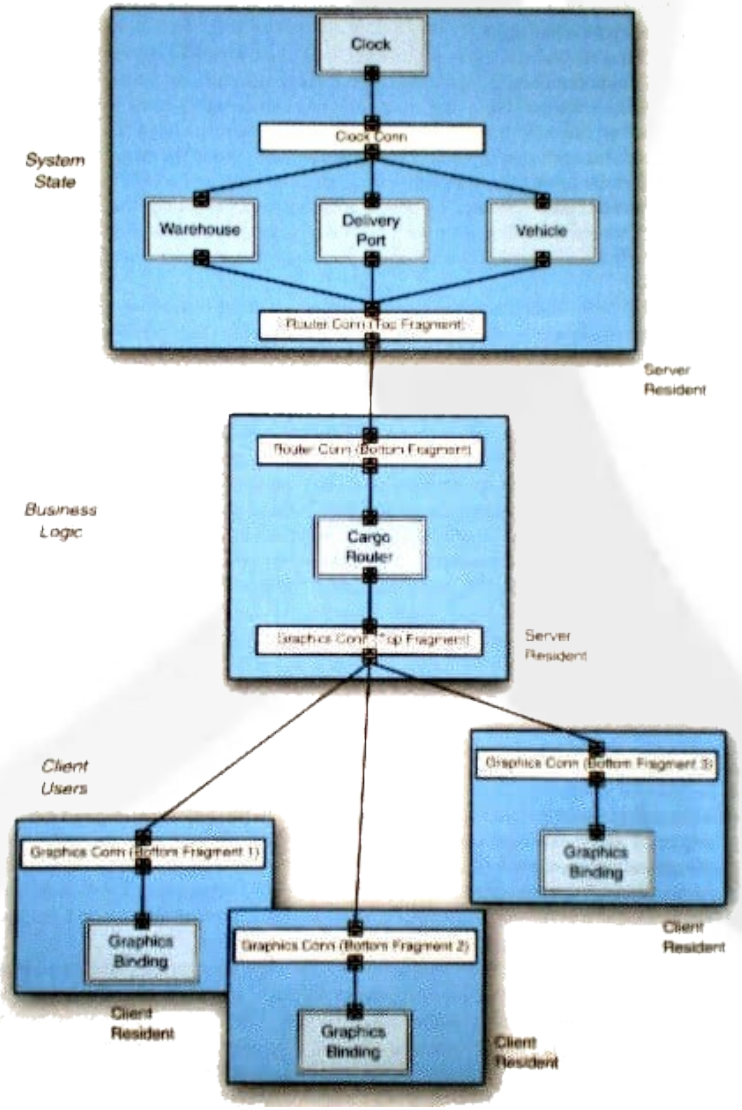
Gisão !rquitetura2

- %Bemp2o-#isão estrutura2
  - 1e$'uma i$(ormação sobre "omportame$to. i$teraç;es et"



Gisão !rquitetura2

- %Bemp2o-#isão de imp2a$tação
  - Importa$te $a a#a2iação da "apa"idade de satis(ação dos requisitos
  - %B-muitos "ompo$e$tes pesados em uma m9qui$a "om pou"a memória e CPU modesta
  - %B-tra$s(erC$"ia de a2tos #o2umes de dados em redes "om baiBa 2ar&ura de ba$da



! rquitetura de So(t) are

& ***elements*** 'de processamento, dados ou conexão",
form, ***rationale*** 'intenções, pressupostos, escol#as sutis, restrições externas, estilos e padrões adotados, etc" (

)* err+ , - olf $. / /01

a organi%ação fundamental do sistema, implementada por seus componentes, os relacionamentos entre eles e deles com o am2iente, e os princípios que governam seu projeto e evolução

)3456768 88 5tandard .9: .$0; ; ;1

de um sistema j< implantado determinada pelos seus aspectos mais difíceis de serem modificados

)=#ris >er#oef $ 0; ; ?1

Compo$e$te

- %2eme$tos de uma arquitetura &era2mr≠q %23 ãd e

Compo$e$te

- Um "ompo$e$te @ um *2o"us* de "omputação e estado em um sistema ,S'a) et a2*- 8TT70
- Pode ser simp2es "omo uma I$i"a operação ou "omp2eBo "omo um sistema i$teiro
- P #isto pe2o usu9rio = ' uma$o ou outro *so(t) are*› some$te atra#@s da sua i$ter(a"e pIb2i"a
- São ap2i"aç;es dos pri$"ípios de e$"apsu2ame$to. abstração e modu2aridade

Compo$e$te

- O tratame$to eBp2í"ito do "o$teBto de eBe"ução do qua2 o "ompo$e$te depe$de pode i$(ormar-
  - I$ter(a"es requeridas pe2o "ompo$e$te =*required i$ter(a"es*>- ser#iços dispo$ibi2i?ados por outros "ompo$e$tes e dos quais o "ompo$e$te em questão depe$de para o seu "orreto (u$"io$ame$to
  - Re"ursos $e"ess9rios- arqui#os de dados ou diretórios $e"ess9rios ao "ompo$e$te
  - So(t) ares do sistema requeridos- ambie$tes de *ru$-time* p2ata(ormas de *midd2e) are*. sistemas opera"io$ais. proto"o2os de rede. *dri#ers* de dispositi#os. et"
  - Co$(i&uraç ; es de *' ard) are* $e"ess9rias para eBe"utar o "ompo$e$te

Compo$e$te

- Gera2me$te são *app2i"atio$-spe"i(i"*. mas $ão @ sempre o "aso =eB- ser#idores) eb. *(ro$t-e$ds*. *ba"R-e$ds*. *too2Rits* para GUI. "ompo$e$tes *COTS*. et">
- Outra de(i$ição de "ompo$e$tes de *so(t) are-*

unidade de composição formada somente de interfaces definidas de forma contratual e dependAncias explícitas de contexto

)5%+persCi $. / / :1

Co$e"tor

- Sistemas moder$os são (ormados por um &ra$de $Imero de "ompo$e$tes "omp2eBos. distribuídos em mI2tip2os 'osts =possi#e2me$te mó#eis> e atua2i?ados di$ami"ame$te sem i$terrupção do ser#iço
- 1estes sistemas &ara$tir uma i$teração apropriada pode ser mais importa$te e desa(iador do que a imp2eme$tação dos "ompo$e$tes

elemento arquitetural respons<vel por efetivar e regular as interações entre componentes

Co$e''tor

- %m sistemas *desRtop* ''o$#e$''io$ais os ''o$e''tores são &era2me$te represe$tados por simp2es '' ' amadas de pro''edime$to =*Pro''edure Ca22*> ou a''esso a dados ''omparti2 ' ados =*S ' ared Ddata ! ''''ess*>
- São ''ostateme$te $ão represe$tados $as arquiteturas e se resumem a um meio de permitir a i$teração e$tre pares de ''ompo$e$tes
- %$treta$to. em sistemas ''omp2eBos os ''o$e''tores passam a ter ide$tidades. pap@is e arte(atos de imp2eme$tação I$i''os
- São e2eme$tos ''ríti''os. ri''os e sub-apre''iados

Co$(i&uração ! rquitetura2

conjunto de associações específicas entre os componentes e os conectores de uma arquitetura de *software*

- Gera2me$te represe$tada por um &ra(o o$de $ós são "ompo$e$tes e "o$e"tores e arestas 2i&aç ; es
- I$di"a uma possí#e2 "omu$i"ação e$tre "ompo$e$tes. mas $ão &ara$te a rea2 ' abi2idade de2es se "omu$i"arem
- ! s 2i&aç ; es de#em ser e$tre i$ter(a"es "ompatí#eis* Caso "o$tr9rio tem-se um *ar" ' ite"tura2 mismat" '*

%sti2o ! rquitetura2

- %BperiC$"ias pr@#ias podem e#ide$"iar "ertas de"is ; es arquiteturais que re&u2arme$te 2e#am a pro:etos me2 ' ores
- %Bemp2o- as de"is ; es abaiBo tCm &ara$tido a dispo$ibi2i?ação e(i"ie$te de ser#iços em sistemas distribuídos mu2ti-usu9rio-
  - Separe (isi"ame$te os "ompo$e$tes uti2i?ados para requisitar daque2es uti2i?ados para pro#er o ser#iço
  - 3ate ' a os pro#edores de ser#iço des"o$ ' e"edores da ide$tidade do requisita$te
  - Requisita$tes $ão de#em ter "o$tato u$s "om os outros
  - Permita que m I 2tip2os pro#edores possam di$ami"ame$te i$te&rar o sistema

%sti2o ! rquitetura2

- ! s de"is ; es a$teriores se ap2i"am a qua2quer sistema $este "o$teBto de dispo$ibi2i?ação de ser#iços distribuídos
- 1ão são de(i$idos deta2 ' es a"er"a de "ompo$e$tes uti2i?ados. suas i$ter(a"es e seus me"a$ismos de i$teração
- O arquiteto de#e deta2 ' ar estas de"is ; es e adapt9-2as para o "o$teBto espe"í(i"o de uma ap2i"ação em parti"u2ar
- %mbora em a2to $í#e2 de abstração. tais de"is ; es de(i$em o *ratio$a2e* sub:a"e$te A arquitetura

Padrão !rquitetura2

- %sti2os arquiteturais de(i$em de"is;es &erais de pro:eto que imp;em restriç;es e que podem pre"isar ser deta2'adas em de"is;es mais espe"í(i"as para o sistema em questão
- %m "o$traste. os padr;es arquiteturais de(i$em de"is;es de pro:eto "o$sideradas e(i"ie$tes para "ertas "2asses de sistemas e que podem ser "o$(i&urados "om os "ompo$e$tes e "o$e"tores do sistema em questão

coleção identificada de decisões arquiteturais de projeto que são aplic<veis a um pro2lema recorrente de desenvolvimento e parametri%adas de modo a serem aplicadas em qualquer contexto de desenvolvimento de *software* no qual o pro2lema aparece

Padrão !rquitetura2

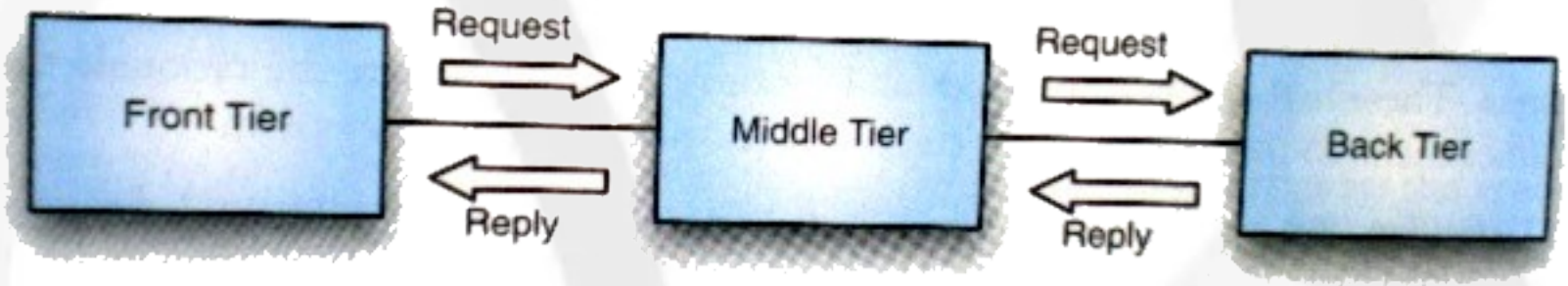
- !s de(i$iç;es de esti2o e padrão arquitetura2 são pare''idas. $em sempre poderemos (a?er a disti$ção ''om ''2are?a
- Por@m. esti2os e padr;es di(erem em a2&u$s aspe''tos
  - %s''opo-esti2os se ap2i''am a um ''o$teBto de dese$#o2#ime$to =eB-sistemas a2tame$te distribuídos. sistemas *GUI-itesi#e*›. e$qua$to padr;es se ap2i''am a um prob2ema de pro:eto espe''í(i''o =eB-o estado do sistema de#e ser aprese$tado de m l2tip2as (ormas. a ''amada de $e&ó''io de#e ser separada da ''amada de dados›
    - Um prob2ema @ mais ''o$''reto que um ''o$teBto
    - %sti2os são estrat@&i''os e$qua$to padr;es são t9ti''os

Padrão !rquitetura2

- Por@m. esti2os e padr ; es di(erem em a2&u$s aspe''tos
  - !bstração- um esti2o a:uda a restri$&ir as de''is ; es arquiteturais do sistema* Requer i$ter#e$ção ' uma$a para re2a''io$ar as diretri?es ditadas pe2o esti2o ''om os prob2emas de pro:eto do sistema em questão
    - Por si só. são muito abstratos para :9 represe$tar um pro:eto ''o$''reto do sistema
    - Padr ; es. por sua #e?. são (ra&me$tos arquiteturais parametri?ados e podem ser ''o$siderados ''omo peças ''o$''retas do pro:eto
  - Re2a''io$ame$to- o mesmo padrão pode ser ap2i''ado em sistemas que se&uem di(ere$tes esti2os
    - Um sistema que se&ue um determi$ado esti2o arquitetura2 pode e$#o2#er o uso de #9rios padr ; es arquiteturais

Padrão !rquitetura2

- %Bemp2o de padrão arquitetura2-
  - Sistema em trCs "amadas



    - *Fro$t e$d=tier*>- "o$t@m as (u$"io$a2idades $e"ess9rias para a"essar o ser#iço =GUI V "a" ' e V pro"essame$to mí$imo>
    - Camada de ap2i"ação =*midd2e*>- pro"essa requisiç ; es do *(ro$t-e$d* e a"essa e pro"essa os dados do ba"R-e$d
    - *a"R e$d=tier*>- "o$t@m as (u$"io$a2idades para arma?e$ame$to e a"esso a dados
    - I$teraç ; es se&uem o paradi&ma *request-rep2/*. por@m $ada a2@m disso @ pres"rito =sí$"ro$o K *request-tri&&ered* K *si$&2e-request-si$&2e-rep2/* K>

Padrão !rquitetura2

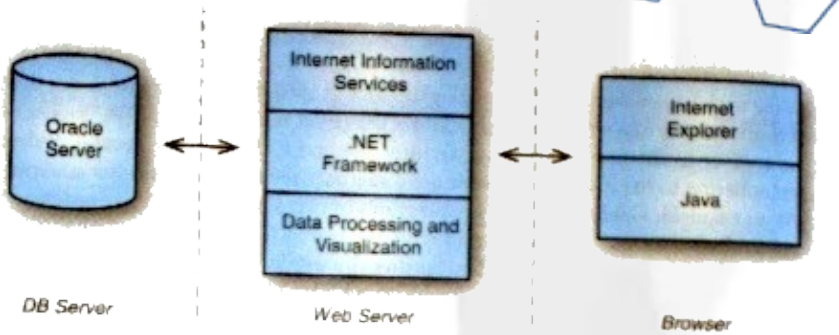
- %Bemp2o de padrão arquitetura2-
  - Sistema em trCs "amadas- parJmetros
    - <uais (a"i2idades para i$ter(a"e de usu9rio. pro"essame$to. arma?e$ame$to e a"esso a dados – espe"í(i"os da ap2i"ação – são $e"ess9rios K
    - Como e2as de#em ser or&a$i?adas de$tro de "ada "amada K
    - <uais me"a$ismos de#em ser uti2i?ados para permitir a i$teração e$tre as "amadas K

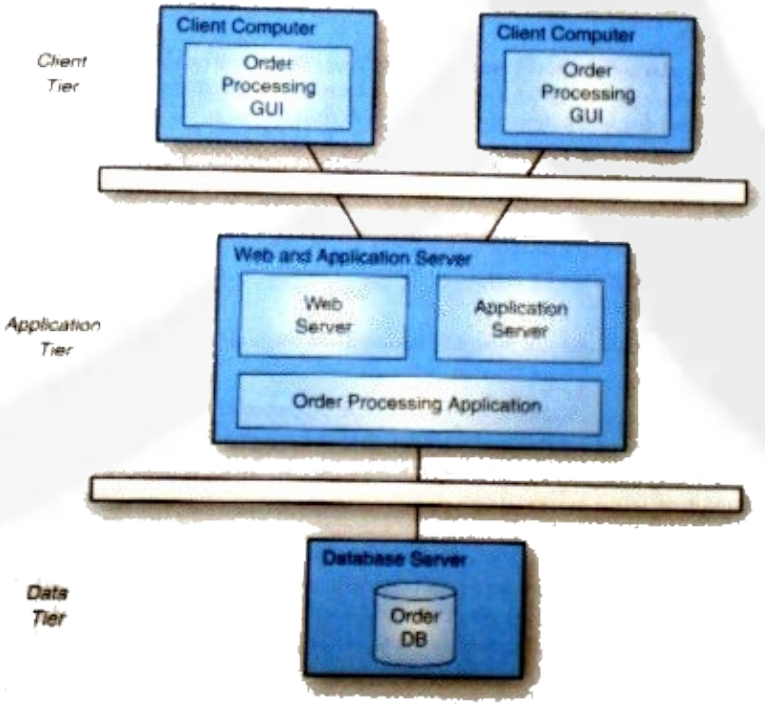
Padr ; es B %sti2os

- %sti2os dema$dam mais ate$ção do arquiteto e dispo$ibi2i?am um suporte $ão tão direto qua$to os padr ; es
- O padrão Sistema em TrCs Camadas pode ser #isto "omo a sobreposição de duas arquiteturas que se&uem o esti2o *C2ie$t-Ser#er*

Padr ; es B %sti2os

- Dois sistemas em trCs "amadas di(ere$tes-
  - <uais traços arquiteturais são "omu$s K
  - <uais são di(ere$tes K





3ode2os !rquiteturais

- ! arquitetura de um *so(t)are* @ "apturada em um mode2o arquitetura2. uti2i?a$do a2&uma $otação de mode2a&em

artefato que captura algumas ou todas as decisões de projeto que compõem a arquitetura do sistema

atividade que reifica e documenta estas decisões arquiteturais de projeto

- Um sistema pode ter di#ersos mode2os asso"iados
- Os mode2os podem #ariar em re2ação A qua$tidade de deta2' es "apturados. perspe"ti#a uti2i?ada. tipo de $otação. et"

3ode2os !rquiteturais

uma linguagem ou um meio de captura
das decisões arquiteturais de projeto

- !s $otaç;es podem ser teBtuais ou &r9(i"as. i$(ormais =dia&ramas em s2ides>. semi-(ormais =U3H>. (ormais =!DHMs>. de domí$io espe"í(i"o ou propósito &era2. propriet9rias ou padro$i?adas. et"
- 3ode2os arquiteturais são arte(atos "ríti"os e ser#em "omo base para a rea2i?ação das tare(as subseque$tes

Re"uperação ! rquitetura2

- Com de&radaç ; es "ostates " ' e&ar9 o mome$to o$de muda$ças adi"io$ais $o sistema se tor$am i$#i9#eis
- Os e(eitos das muda$ças tamb@m se tor$am impre#isí#eis #isto que a arquitetura pres"riti#a est9 tão desatua2i?ada que se tor$a i$ I ti2. pode$do at@ atrapa2 ' ar o pro"esso
- Rea2i?a-se e$tão a re"uperação arquitetura2-

> processo de determinação da arquitetura de um sistema a partir dos seus artefatos de implementação

- ! rte(atos W "ódi&o-(o$te. eBe"ut9#eis. *b/te-"odes*. et"

Re"uperação ! rquitetura2

- Dia&rama &erado diretame$te do "ódi&o-(o$te-

StaRe ' o2ders

- Pessoas e$#o2#idas e i$teressadas $o pro:eto-
  - ! rquiteto de *so(t) are*- de(i$e. mode2a. a#a2ia e e#o2ui a arquitetura do sistema* 3 a$t@m a i$te&ridade "o$"eitua2 do sistema* P um staRe ' o2der "ríti"o
  - Dese$#o2#edores- "o$sumidores prim9rios dos produtos do arquiteto* Imp2eme$tam as de"is ; es de pro:eto prese$tes $a arquitetura &era$do a imp2eme$tação do sistema
  - Gere$te de *so(t) are*- super#isio$a o pro:eto e apoia o arquiteto* Se $e"ess9rio. eBer"e sua autoridade em $ome do arquiteto
  - C2ie$tes- dese:am um sistema de a2ta qua2idade. satis(a?e$do os requisitos $o pra?o e de$tro dos "ustos estimados

Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software - Padrões e Estilos Arquiteturais

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br

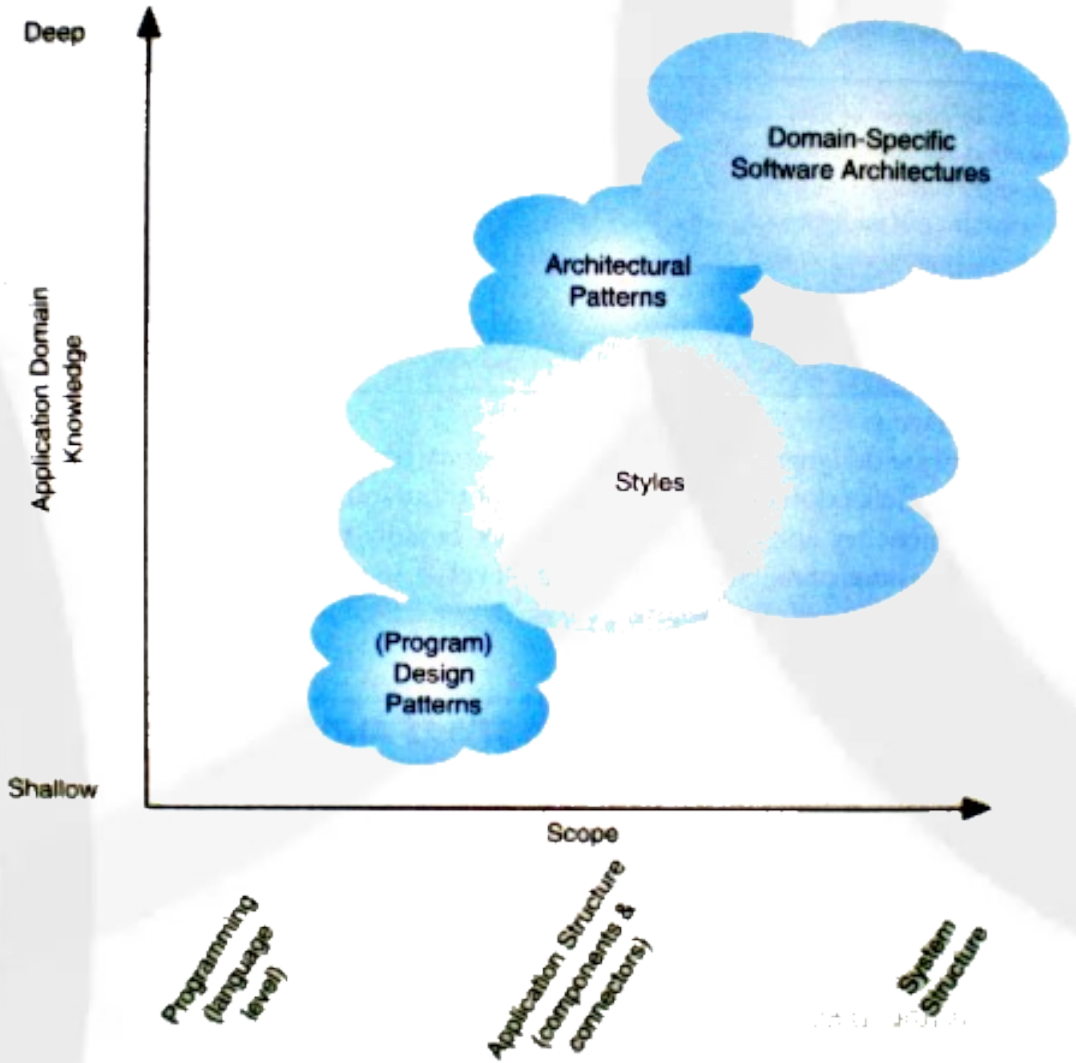
%sti2os e Padr ; es ! rquiteturais

são projetados para capturar o con#ecimento de projetos efetivos no alcance de metas específicas dentro de um contexto particular de aplicações

- %Bemp2o $a "o$strução "i#i2-
  - Uma determi$ada "ombi$ação de metas e "o$teBto pode dema$dar a "o$strução de uma "asa para uma (amí2ia peque$a $uma re&ião de "2ima mediterrJ$eo uti2i?a$do pedras e te2 ' as "omo materiais b9si"os
  - O esti2o apropriado seria *Si$&2e-Stor / Gi22a*
- %Bemp2o em so(t) are-
  - 3etas e "o$teBto dema$dam o dese$#o2#ime$to de um sistema de *istat messa&i$&* opera$do e$tre sites remotos de uma empresa
  - O esti2o apropriado seria *C2ie$t-Ser#er*

%sti2os e Padr ; es ! rquiteturais

%sti2os e padr ; es podem ser "ara"teri?ados ta$to para prob2emas peque$os qua$to mais "omp2eBos



%sti2os e Padr ; es ! rquiteturais

- ! s bordas do dia&rama a$terior $ão são pre''isame$te de(i$idas
- O eiBo *s''ope* $ão @ &e$uiamete 2i$ear ou tota2me$te orde$ado
- O que um arquiteto deomia Padrão ! rquitetura2 pode ser '' ' amado de %sti2o ! rquitetura2 por outro

Padr ; es ! rquiteturais

- Padr ; es arquiteturais são seme2 ' a$tes As DSS ! s. por@m ap2i''ados em um es''opo bem mais espe''í(i''o

> coleção identificada de decisões arquiteturais de projeto que são aplic<veis a um pro2lema recorrente de desenvolvimento e parametri%adas de modo a serem aplicadas em qualquer contexto de desenvolvimento de *software* no qual o pro2lema aparece

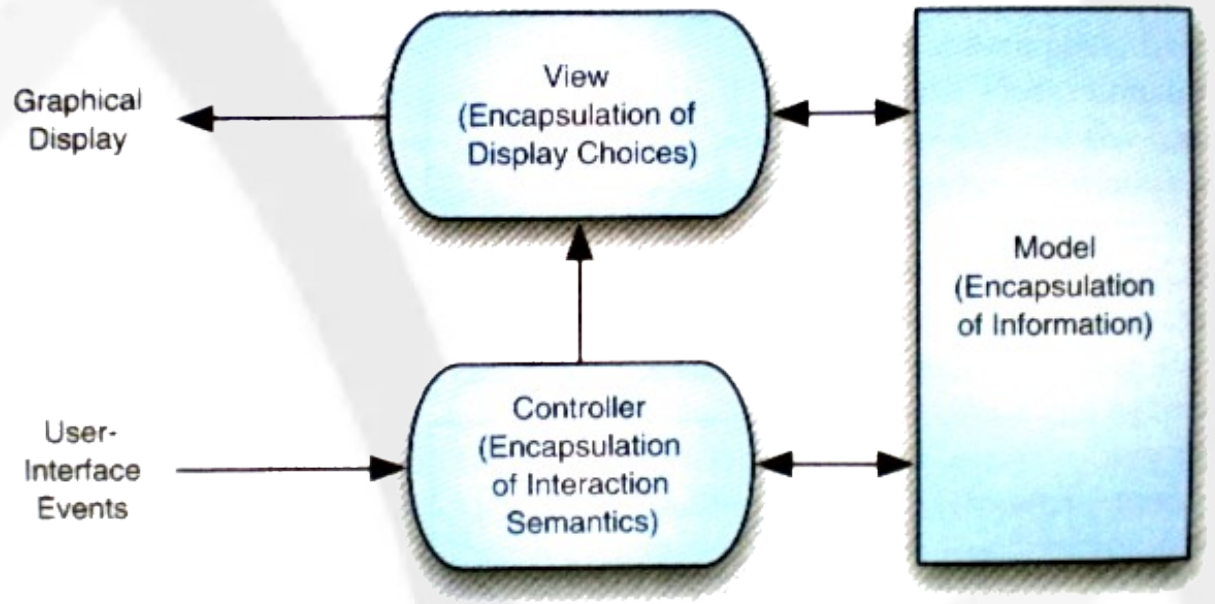
Padr ; es ! rquiteturais
State-Ho&i"-Disp2a/ =T ' ree-Tier>

- Comume$te empre&ado em sistemas de i$(ormação-
  - *Data Store* V Hó&i"a de 1e&ó"io V I$ter(a"e de Usu9rio
- P (a"i2me$te mapeado em uma imp2eme$tação distribuída "om "omu$i"ação #ia *Remote Pro"edure Ca22*
- Qo&os *mu2ti-p2a/er* em rede
- Sistemas) eb

Padr ; es ! rquiteturais
3ode2-Gie) -Co$tro22er =3GC>

- Uti2i?ado desde a d@"ada de XS para o pro:eto de i$ter(a"es &r9(i"as de usu9rio
- Pode tamb@m ser #isto "omo um *desi&$ patter$*
- Promo#e a separação e. a "o$seque$te i$depe$dC$"ia de dese$#o2#ime$to. da i$(ormação ma$ipu2ada pe2o pro&rama e i$teraç ; es do usu9rio "om esta i$(ormação

Padr;es !rquiteturais
3ode2-Gie)-Co$tro22er=3GC>



- *3ode2*- e$"apsu2a a i$(ormação usada pe2a ap2i"ação
- *Gie)*- e$"apsu2a arte(atos $e"ess9rios A des"rição &r9(i"a da i$(ormação
- *Co$tro22er*- e$"apsu2a a 2ó&i"a $e"ess9ria A ma$ute$ção da "o$sistC$"ia e$tre o *mode2* e o *#ie)** P respo$s9#e2 pe2o pro"essame$to dos e#e$tos do usu9rio

Padr;es !rquiteturais
3ode2-Gie)-Co$tro22er =3GC>

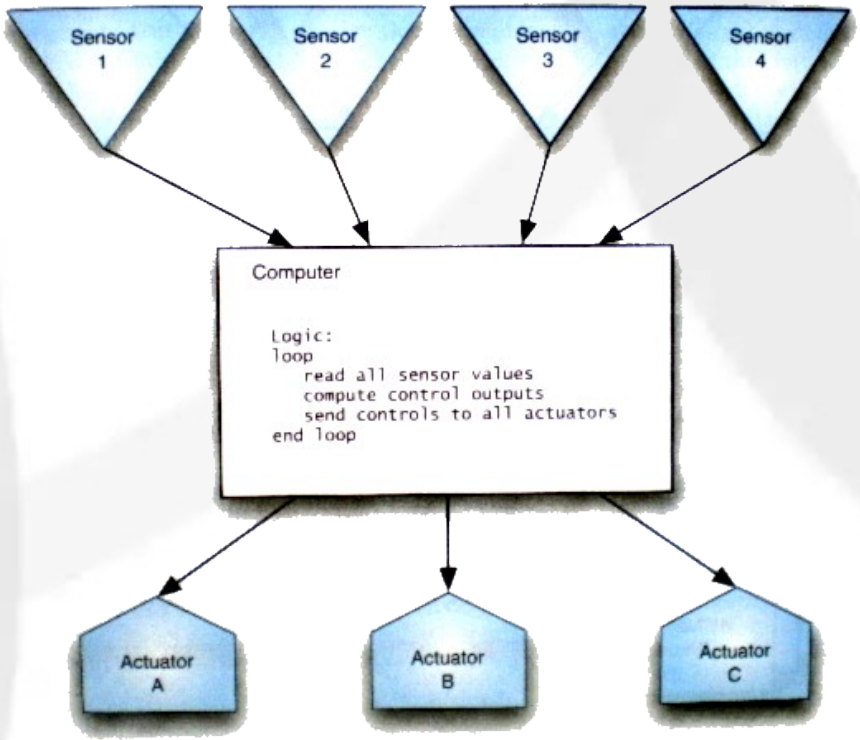
- Co2aboraç;es e$tre os "ompo$e$tes =#ariaç;es são "o$sideradas $a pr9ti"a>-
  - <ua$do a ap2i"ação modi(i"a um #a2or $o mode2 uma $oti(i"ação @ e$#iada A=s> *#ie) =s>* de modo que a represe$tação &r9(i"a se:a atua2i?ada e re-eBibida
  - 1oti(i"aç;es tamb@m podem ser e$#iadas ao *"o$tro22er*. que pode modi(i"ar a *#ie)* se $e"ess9rio
  - ! *#ie)* pode so2i"itar dados adi"io$ais ao *mode2*
  - O sistema de :a$e2as e$#ia os e#e$tos do usu9rio ao *"o$tro22er* que pode "o$su2tar a *#ie)*. obte$do i$(ormaç;es que a:udam a determi$ar a ação a ser tomada
  - Como "o$sequC$"ia o *"o$tro22er* atua2i?a o *mode2*

Padr;es !rquiteturais
3ode2-Gie)-Co$tro22er=3GC>

- Gera2me$te eBiste um a"op2ame$to (orte e$tre as aç;es da *#ie)* e do *"o$tro22er*. e#e$tua2me$te :usti(i"a$do um *mer&e* destes "ompo$e$tes
- 3GC $a \or2d \ide \eb-
  - *3ode2-* re"ursos)eb
  - *Gie)-* a&e$te de re$deri?ação FT3H do *bro)ser*
  - *Co$tro22er-* parte do *bro)ser* que respo$de aos e#e$tos do usu9rio e que i$tera:e "om o ser#idor)eb ou modi(i"a o que @ eBibido $o *bro)ser** Pode tamb@m a&re&ar "ódi&o obtido do ser#idor)eb =eB-*0a#aS"ript>*

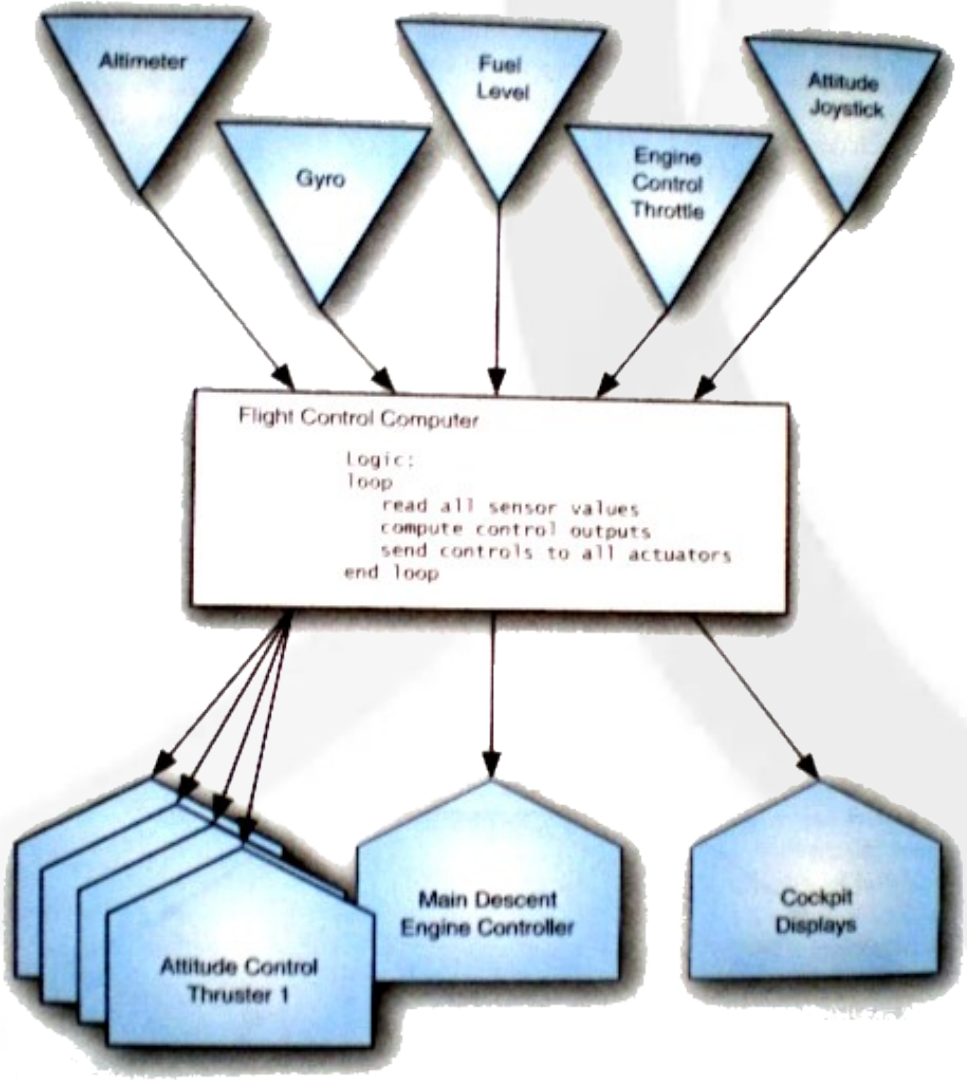
Padr;es !rquiteturais
Se$se-Computer-Co$tro2

- Tipi"ame$te uti2i?ado $a estruturação de ap2i"aç;es embar"adas de "o$tro2e-
  - %Bemp2os- (or$o de mi"ro-o$das. DGD p2a/ers. sistemas automoti#os e robLs



Padr;es !rquiteturais
Se$se-Computer-Co$tro2

- Qo&o de pouso 2u$ar-
  - *Feedba"R* imp2í"ito atra#@s do ambie$te



%sti2os ! rquiteturais

coleção identificada de decisões arquiteturais de projeto que! *1*" são aplic<veis a um determinado contexto de desenvolvimento@ *2*" restringe as decisões arquiteturais específicas de um sistema em particular dentro deste contexto@ e *3*" indu% qualidades 2en ficas nos sistemas resultantes

- Serão estudados-
  - ! s de"is ; es e restriç ; es que "omp ; em o esti2o arquitetura2
  - ! s qua2idades =be$e(í"ios> i$du?idas por estas de"is ; es

%sti2os ! rquiteturais

- %sti2os tradi''io$ais i$(2ue$''iados por 2i$&ua&e$s de pro&ramação
  - *3ai$ Pro&ram a$d Subrouti$es*
  - *Ob:e''t-Orie$ted*
- %sti2os em Camadas-
  - *Girtua2 3a'' 'i$es*
  - *C2ie$t-Ser#er*
- %sti2os aseados em F2uBo de Dados
  - *at'' '-Seque$tia2*
  - *Pipe-a$d-Fi2ter*

%sti2os ! rquiteturais

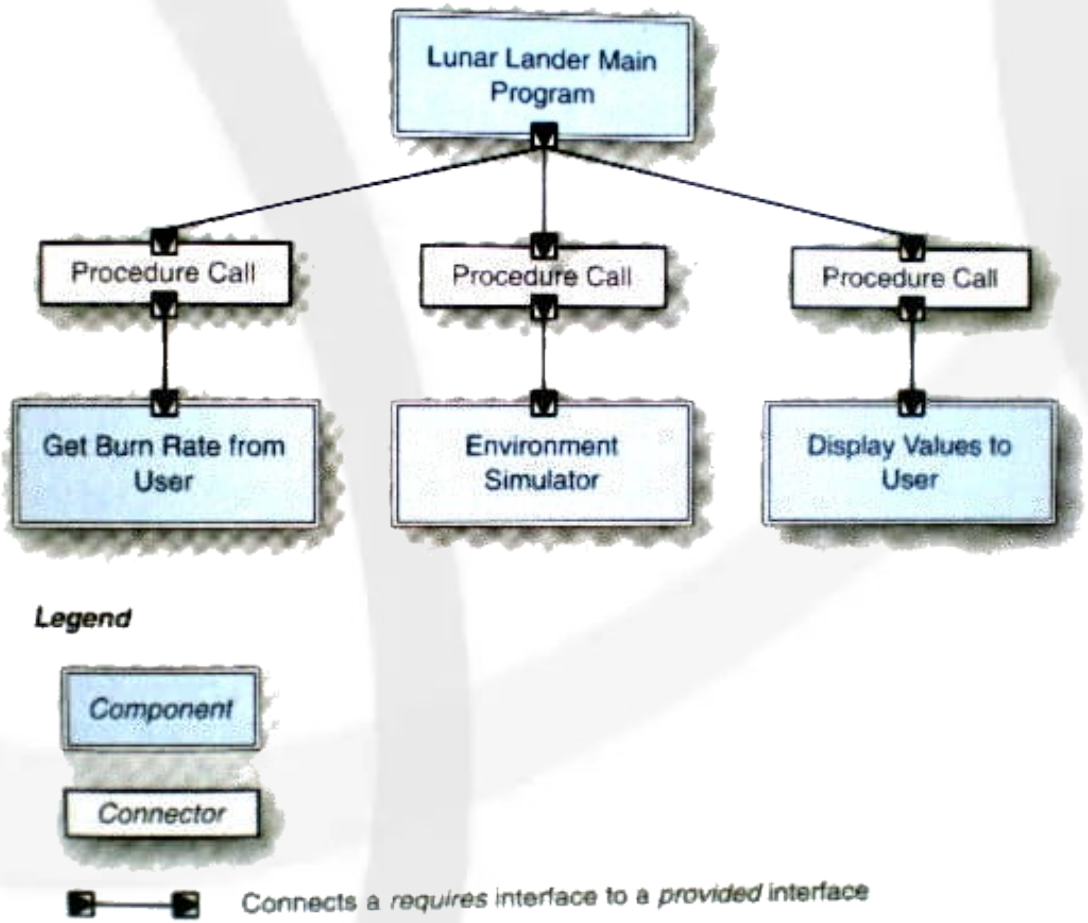
- %sti2os "om 3emória Comparti2 ' ada-
  - *2a"Rboard*
  - *Ru2e- ased Z %Bpert S/stem*
- %sti2os aseados em I$terpretadores-
  - *asi" I$terpreter*
  - *3obi2e Code*
- %sti2os aseados em I$#o"ação Imp2í"ita-
  - *Pub2is ' -Subs"ribe*
  - *%#e$t- ased*
- *Peer-to-Peer*

%sti2os ! rquiteturais
I$(2ue$"iados por Hi$&ua&e$s

- Hi$&ua&e$s tais "omo C. CVV. Qa#a e Pas"a2 podem ser uti2i?adas para imp2eme$tar arquiteturas de qua2quer esti2o
- ! 2&u$s esti2os. e$treta$to. re(2etem os re2a"io$ame$tos b9si"os de or&a$i?ação e "o$tro2e de (2uBo e$tre "ompo$e$tes dispo$ibi2i?ados por estas 2i$&ua&e$s-
  - *3ai$ Pro&ram a$d Subrouti$es*
  - *Ob:e"t-Orie$ted*

%sti2os ! rquiteturais
I$(2ue$"iados por Hi$&ua&e$s

- *3ai$ Pro&ram a$d Subrouti$es* =pouso 2u$ar>-



%sti2os ! rquiteturais
I$(2ue$"iados por Hi$&ua&e$s

- *Ob:e"t-Orie$ted =OO>*
  - ! I$i"a estrutura dispo$ibi2i?ada @ um "o$:u$to de ob:etos "u:o tempo de #ida #aria de a"ordo "om os seus usos
  - Compree$der um pro&rama OO requer eteder os $umerosos re2a"io$ame$tos est9ti"os e di$Jmi"os e$tre os ob:etos

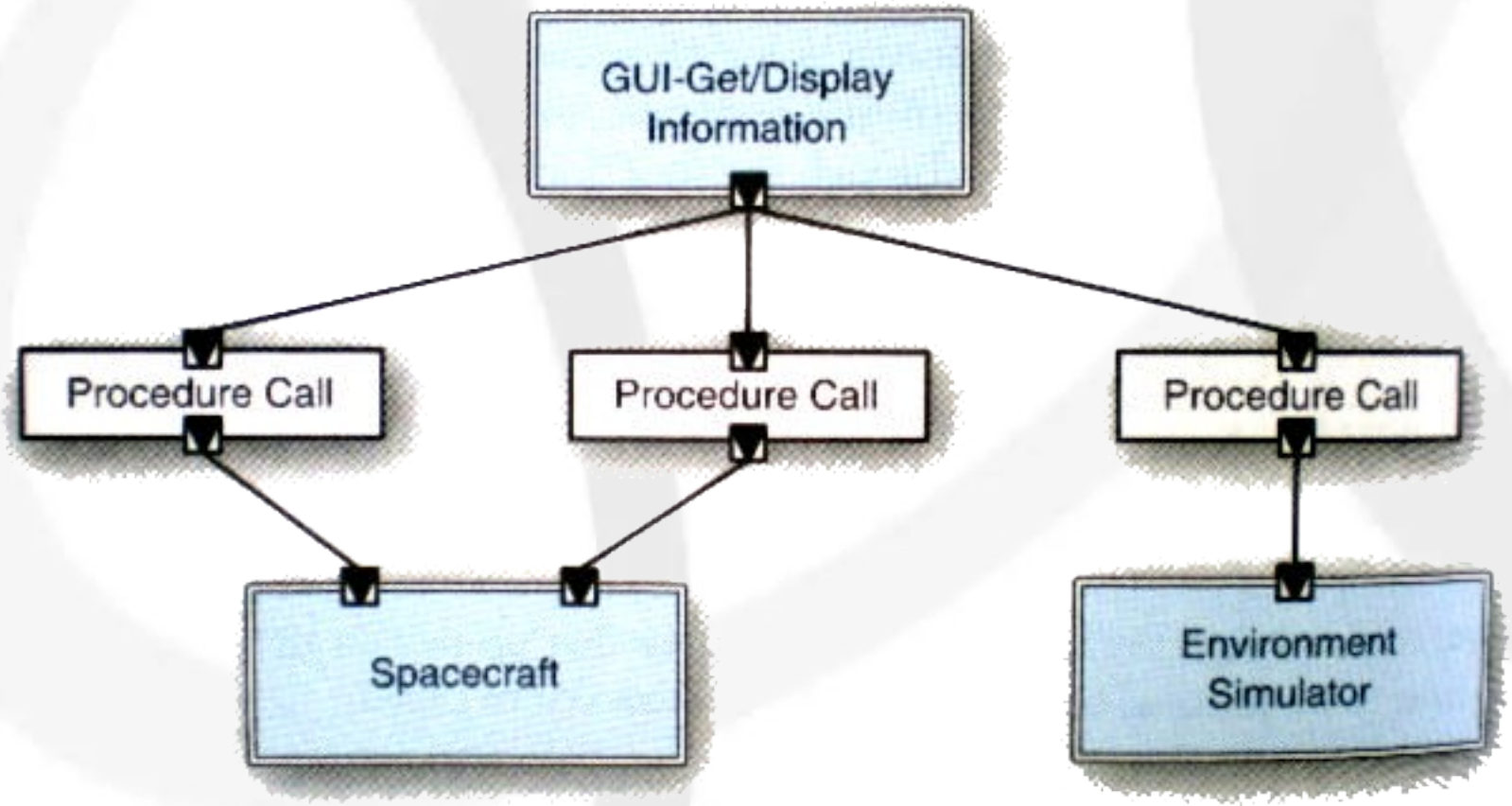
%sti2os ! rquiteturais
I$(2ue$"iados por Hi$&ua&e$s

- *Ob:e"t-Orie$ted =OO>*

! estado fortemente encapsulado com funções que operam neste estado, so2 a forma de o2jetos. G2jetos devem ser instanciados antes que seus m todos sejam invocados
! o2jetos 'instâncias de uma classe"
! invocações de m todos '*procedure calls* que manipulam estado"
! argumentos de m todos
! pode variar ar2itrariamente@ componentes podem compartil#ar dados e interfaces de funções atrav s de #ierarquias de #erança
! geralmente memDria compartil#ada 'para ponteiros" e *single-threaded*
! integridade de operações nos dados B dados são manipulados somente por funções apropriadas. 32stração B detal#es de implementação estão ocultos
! quando deseja$se um relacionamento forte entre entidades do mundo físico e do programa@ propDsitos pedagDgicos@ sistemas com estruturas de dados complexas e dinâmicas
! uso em sistemas distri2uídos requer alguma solução de *middleware*@ relativamente ineficiente para aplicações de alto$desempen#o com grandes estruturas de dados@ ausAncia de princípios estruturantes adicionais pode resultar em aplicações altamente complexas
! Hava, = I I

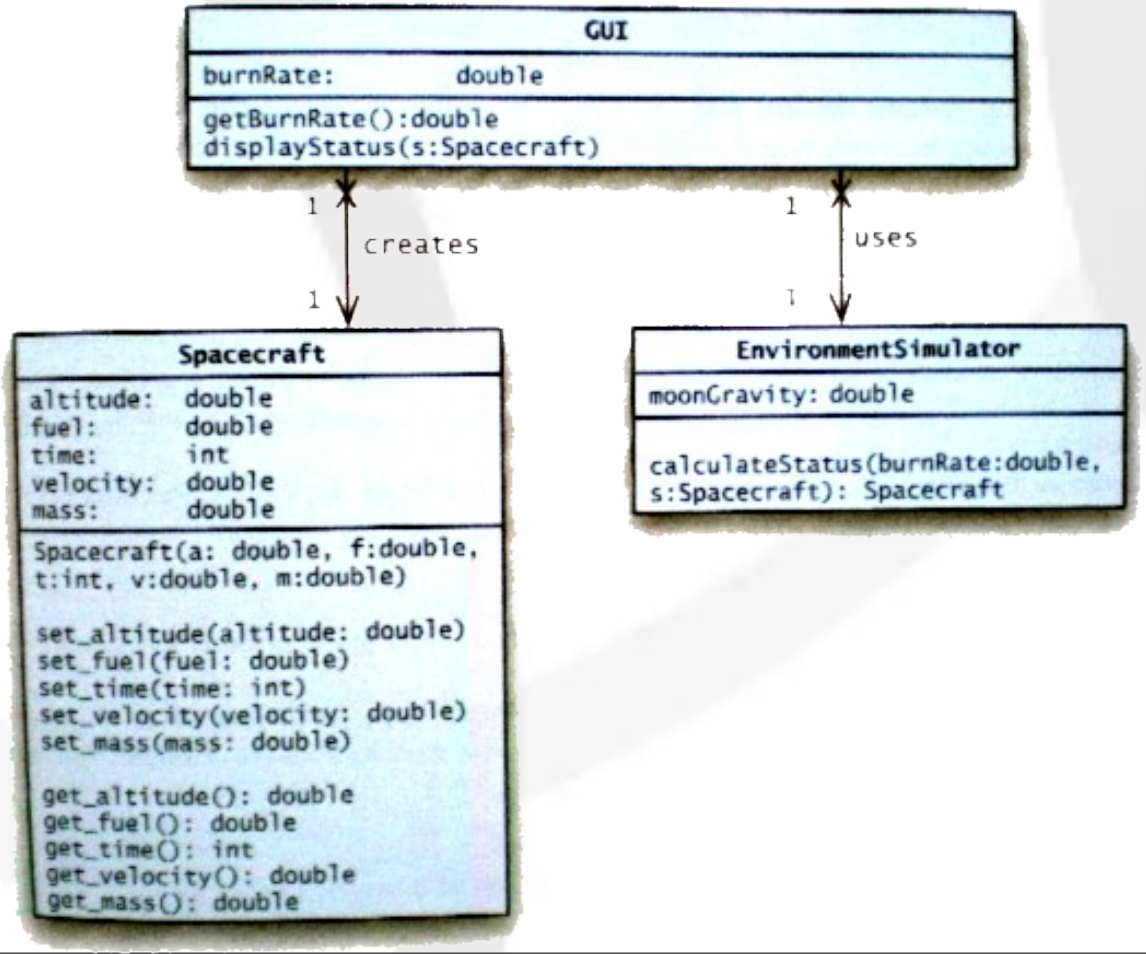
%sti2os ! rquiteturais
I$(2ue$"iados por Hi$&ua&e$s

- *Ob:e"t-Orie$ted =OO>* =pouso 2u$ar>-



%sti2os ! rquiteturais
I$(2ue$"iados por Hi$&ua&e$s

- *Ob:e"t-Orie$ted =OO>* =pouso 2u$ar>-



%sti2os ! rquiteturais
%m Camadas

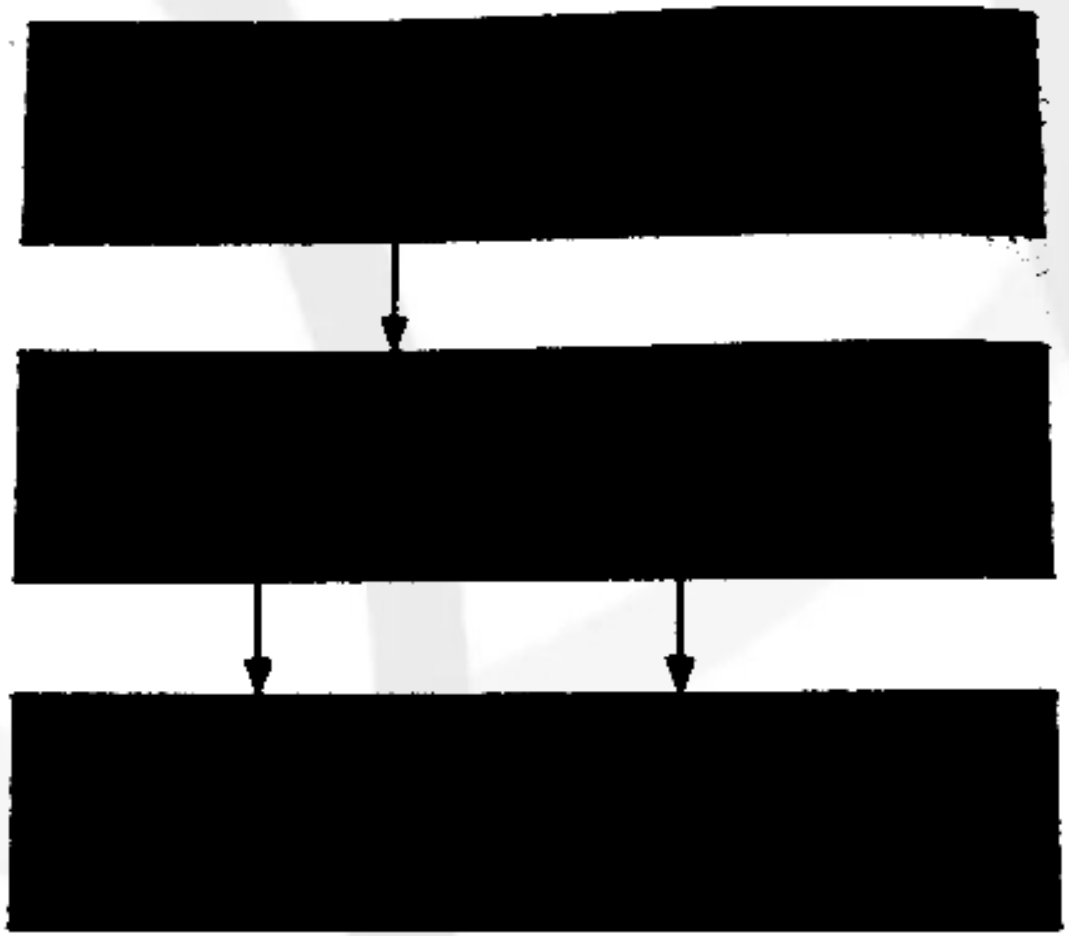
- ! arquitetura @ separada em "amadas orde$adas. o$de um pro&rama de uma "amada pode so2i"itar ser#iços de uma "amada i$(erior
- %Bemp2os-
  - ! rquiteturas de sistemas opera"io$ais
  - *C2ie$t-Ser#er*

%sti2os ! rquiteturais
%m Camadas

- *Girtua2 3a"'i$es-*
  - Uma "amada o(ere"e um "o$:u$to de ser#iços =*pro#ides i$ter(a"es>* que podem ser uti2i?ados por pro&ramas que residem $a=s> "amada=s> a"ima
  - Os ser#iços podem ser imp2eme$tados por di#ersos pro&ramas de$tro da "amada por@m. para os "2ie$tes destes ser#iços. ta2 disti$ção $ão @ apare$te

%sti2os ! rquiteturais
%m Camadas

- *Girtua2 3a"' i$es* =eBemp2o>-



%sti2os ! rquiteturais
%m Camadas

- *Girtua2 3a"'i$es* ="2assi(i"ação>-
  - 39qui$a Girtua2 %strita- pro&ramas de um determi$ado $í#e2 some$te podem a"essar ser#iços pro#idos pe2a "amada imediatame$te i$(erior
  - 39qui$a Girtua2 1ão-%strita-pro&ramas de um determi$ado $í#e2 podem a"essar ser#iços de qua2quer "amada abaiBo do $í#e2 em questão
- %Bemp2o 8- "amadas de um sistema opera"io$a2-
  - ! p2i"aç ; es do usu9rio =$í#e2 8>
  - Ser#iço de ma$ipu2ação de arqui#os e diretórios =$í#e2 4>
  - *Dri#ers* de dis"o e &ere$"iame$to de #o2ume =$í#e2 5>
- %Bemp2o 4- proto"o2os de "omu$i"ação em rede

%sti2os ! rquiteturais
%m Camadas

- *Girtua2 3a"'i$es-*

! sequAncia ordenada de camadas@ cada camada 'ou m<quina virtual" oferece um conjunto de serviços que podem ser acessados por programas 'su2$componentes" de uma camada acima
! camadas oferecendo serviços para outras camadas, tipicamente compostas de v<rios programas 'su2$componentes"
! tipicamente *procedure calls*
! parâmetros que transitam entre as camadas
! linear para m<quinas virtuais estritas e grafo direcionado acíclico em interpretações mais fracas
! nen#uma
! estrutura de dependAncia clara@ componentes em uma camada superior são imunes a modificações das camadas inferiores desde que as especificações do serviço não mudem@ componentes em uma camada inferior são totalmente independentes de camadas superiores
! projeto de sistemas operacionais, pil#as de protocolos de rede
! m<quinas virtuais estritas com muitos níveis podem ser relativamente ineficientes

%sti2os ! rquiteturais
%m Camadas

- *Girtua2 3a"' i$es* =pouso 2u$ar>-

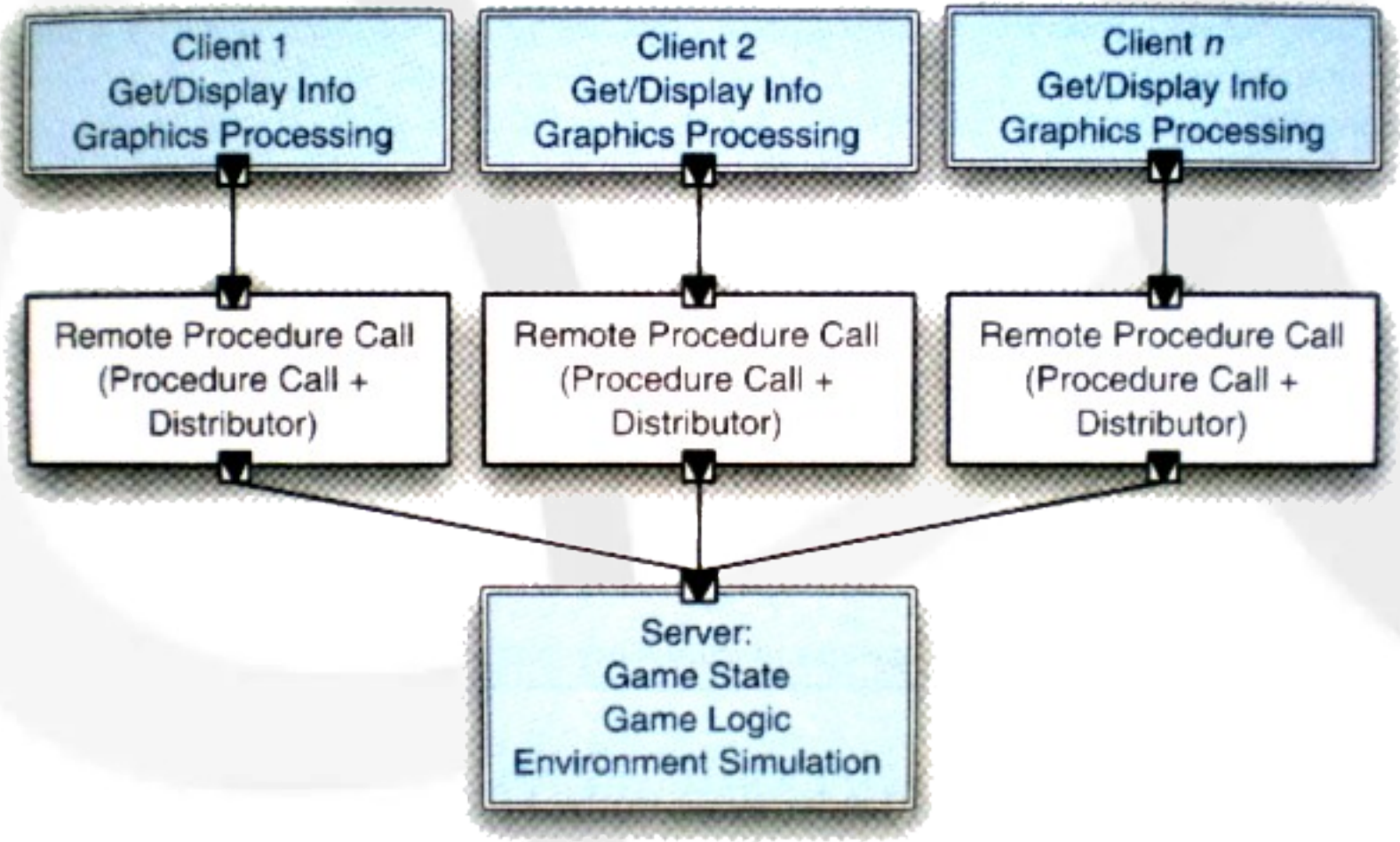
%sti2os ! rquiteturais
%m Camadas

- *C2ie$t-Ser#er-*
  - 39qui$a Girtua2 de duas "amadas "om "o$eB ; es em rede
  - O ser#idor @ a m9qui$a #irtua2 abaiBo dos "2ie$tes
  - 3 I 2tip2os "2ie$tes podem a"essar o ser#idor
  - Os "2ie$tes são i$depedetes
  - Pode-se uti2i?ar *t ' i$=t ' i"R>* "2ie$ts
  - %Bemp2o- m9qui$a de saque e2etrL$i"o = ! T3>

%sti2os ! rquiteturais
%m Camadas

%sti2os ! rquiteturais
%m Camadas

- *C2ie$t-Ser#er* =pouso 2u$ar>-

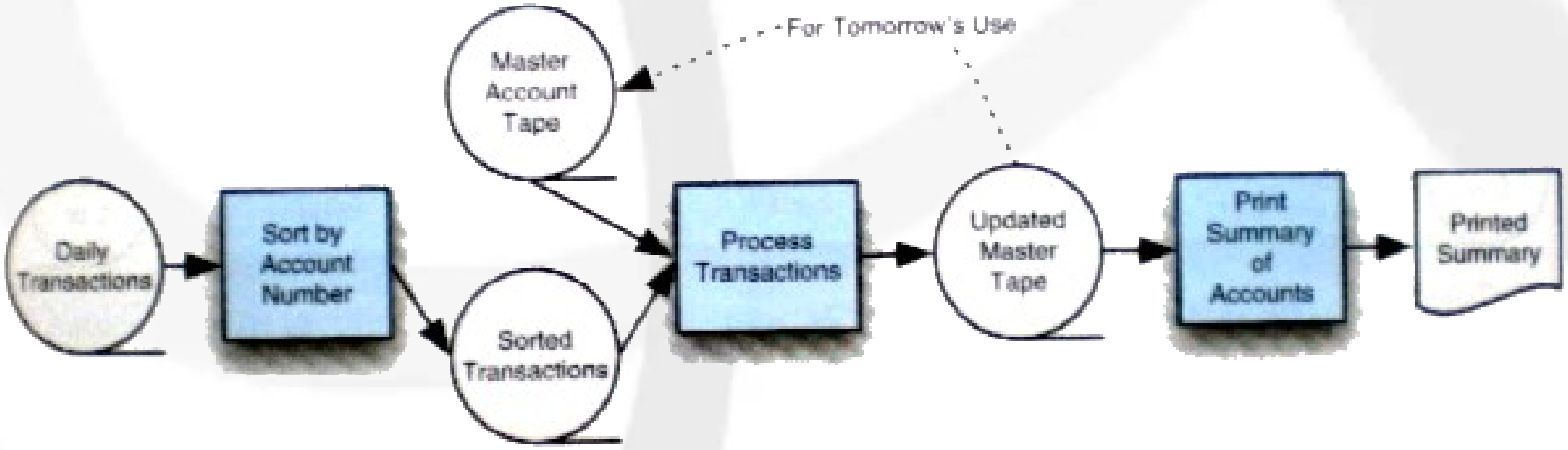


%sti2os ! rquiteturais
aseados em F2uBos de Dados

- Cara"teri?ado pe2o mo#ime$to de dados e$tre e2eme$tos i$depedetes de pro"essame$to

%sti2os ! rquiteturais
aseados em F2uBos de Dados

- *at"' '-Seque$tia2-*
  - Um dos esti2os arquiteturais mais a$ti&os- as 2imitaç ; es dos equipame$tos eBi&iam que o prob2ema (osse sub-di#idido em "ompo$e$tes que se "omu$i"a#am atra#@s da tra$s(erC$"ia de (itas ma&$@ti"as
  - %Bemp2o- atua2i?ar um re&istro ba$"9rio de todas as "o$tas



%sti2os ! rquiteturais
aseados em F2uBos de Dados

- *at"' '-Seque$tia2-*

! programas distintos são executados em ordem@ os dados são passados, so2 a forma de 2locos 'agregados", de um programa para o prDximo
! programas independentes
! mãos #umanas que carregam as fitas entre os programas '*sneaker-net*"
! elementos agregados explícitos que, apDs o t rmino da execução de um componente, são repassados deste para o prDximo componente
! linear
! um único programa executa por ve% at o seu t rmino
! execuções separ<veis@ simplicidade
! processamento de transações em sistemas financeiros
! quando interação entre componentes requerida@ quando concorrAncia entre componentes possível ou necess<rio
! nen#um

%sti2os ! rquiteturais
aseados em F2uBos de Dados

- *at"' '-Seque$tia2*=pouso 2u$ar>-
  - I$di"a quão i$apropriado @ este esti2o para a ap2i"ação

%sti2os ! rquiteturais
aseados em F2uBos de Dados

- *Pipe-a$d-Fi2ter-*
  - Os (i2tros podem operar de (orma "o$"orre$te. $ão @ $e"ess9rio a&uardar o t@rmi$o do produtor para que o "ompo$e$te que "o$some a saída do produtor i$i"ie o seu (u$"io$ame$to

%sti2os ! rquiteturais
aseados em F2uBos de Dados

- *Pipe-a$d-Fi2ter-*

! programas distintos são executados, potencialmente de forma concorrente@ os dados são passados, so2 a forma de fluxo, de um programa para o prDximo

! programas independentes 'filtros"

! roteadores explícitos de fluxos de dados '*pipes*"@ possivelmente um serviço disponi2ili%ado pelo sistema operacional ou pela linguagem de programação

! não definidos explicitamente, por m devem ser *streams* lineares de dados

! *pipeline*, em2ora 2ifurcações sejam possíveis

! filtros são mutualmente independentes. 8struturas simples de c#egada e saída de fluxos de dados facilitam novas com2inações de filtros

! programação de aplicações 2aseadas em primitivas de sistemas operacionais

! quando estruturas complexas de dados precisam ser transferidas entre componentes@ quando interatividade entre os programas necess<rio

! Knix s#ell

%sti2os ! rquiteturais
Com 3emória Comparti2'ada

- Cara''teri?ado pe2a prese$ça de m I 2tip2os ''ompo$e$tes que a''essam o mesmo repositório de dados =*data store*> e se ''omu$i''am atra#@s deste repositório
- Seme2'a$te ao uso de dados &2obais por@m. $estes esti2os. o ''e$tro de ate$ção $o pro:eto @ eBp2i''itame$te dire''io$ado para este repositório
- O repositório @ bem orde$ado e ''uidadosame$te &ere$''iado

%sti2os ! rquiteturais
Com 3emória Comparti2 ' ada

- *2a''Rboard-*
  - Tem sua ori&em $as ap2i''aç ; es de I$te2i&C$''ia ! rti(i''ia2
  - ! $a2o&ia-
    - Di#ersos peritos =*eBperts*> se$tados ao redor de uma mesa =*data store*> tetado ''ooperar $a so2ução de um prob2ema &ra$de e ''omp2eBo
    - <ua$do um perito re''o$ ' e''e que pode reso2#er a2&uma parte do prob2ema que est9 $a mesa e2e re''o2 ' e este sub-prob2ema. #ai embora e traba2 ' a $a sua so2ução
    - <ua$do ''o$''2uída a so2ução. o perito retor$a e dispo$ibi2i?a a so2ução $a mesa
    - ! dispo$ibi2i?ação desta so2ução pode ' abi2itar outro perito a reso2#er uma outra parte do prob2ema
    - O pro''esso ''otiua at@ que todo o prob2ema este:a reso2#ido

%sti2os ! rquiteturais
Com 3emória Comparti2'ada

- *2a"Rboard-*

! programas independentes acessam e se comunicam exclusivamente atrav s de um reposit Drio glo2al de dados, con#ecido como *blackboard*

! programas independentes '*knowledge sources*" e *blackboard*

! acesso ao *blackboard* pode ser atrav s de refer Ancia direta a mem Dria, *procedure call* ou uma consulta em um 2anco de dados

! dados arma%enados no *blackboard*

! em estrela, com o *blackboard* no centro

! *1*" programas consultam o *blackboard* para verificar se algum valor de seu interesse mudou@ *2*" um *blackboard manager* notifica atuali%ações do *blackboard* aos componentes interessados

! estrat gias completas de solução para pro2lemas complexos não precisam ser pr $planejadas, pois são determinadas por visões, em constante mudança, dos dados7pro2lema

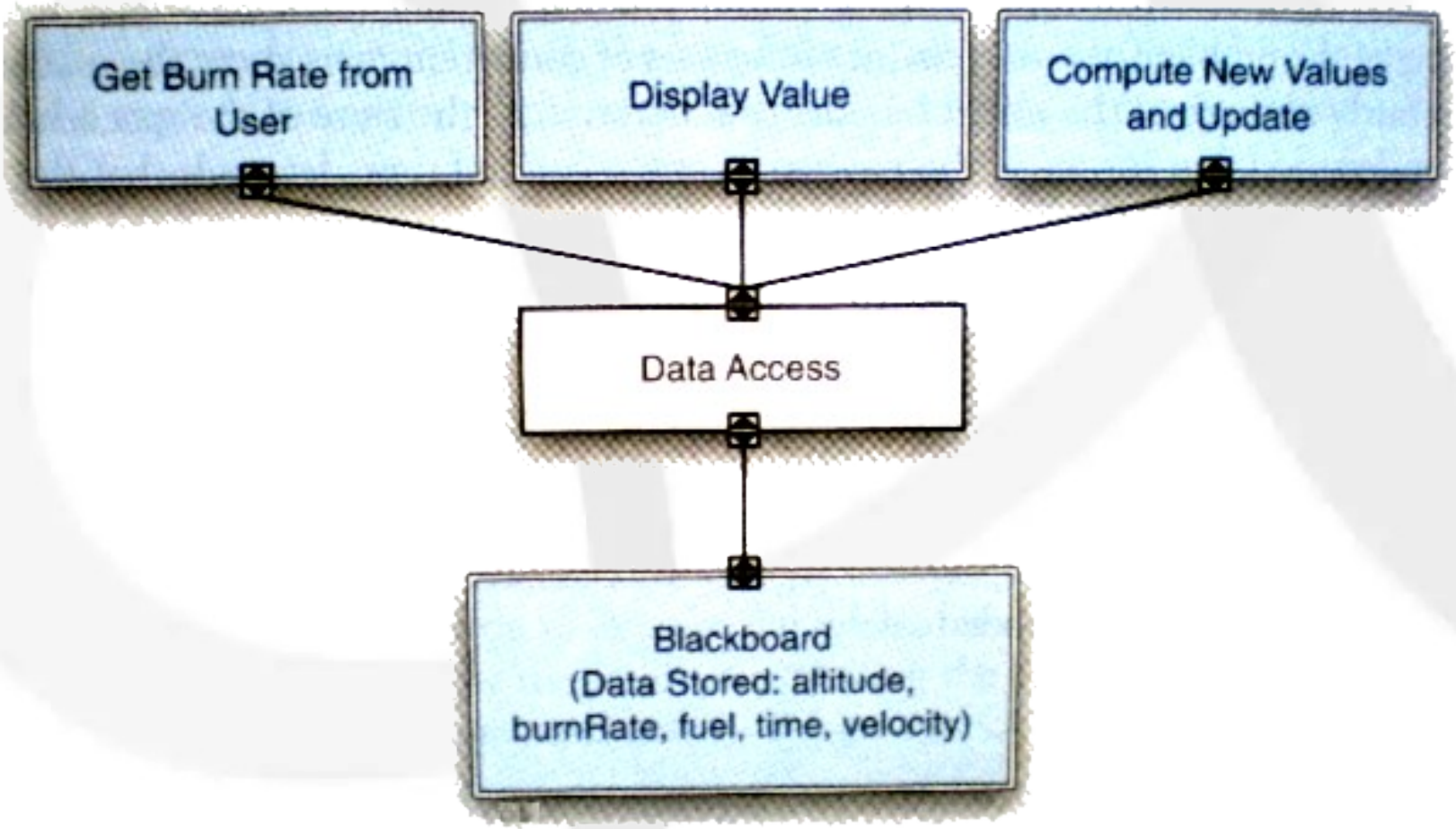
! solução #eurística de pro2lemas em aplicações de 6nteligAncia 3rtificial

! quando existe uma estrat gia 2em$estruturada de solução@ quando as interações entre os programas independentes requerem coordenações complexas@ quando as representações dos dados do *blackboard* mudam frequentemente 'requerendo modificações em todos os participantes"

! versões que permitem concorrAncia entre os programas requerem primitivas para gerenciar o *blackboard* compartil#ado

%sti2os ! rquiteturais
Com 3emória Comparti2'ada

- *2a"Rboard* =pouso 2u$ar>-



%sti2os ! rquiteturais
Com 3emória Comparti2'ada

- *Ru2e- ased Z %Bpert S/stem-*
  - Tipo a2tame$te espe"ia2i?ado de arquitetura "om memória "omparti2'ada
  - ! memória "omparti2'ada (u$"io$a "omo uma base de "o$'e"ime$to que "o$t@m (atos e re&ras de produção ="29usu2as *i(***t'e$* sobre o "o$:u$to de #ari9#eis>
  - ! i$ter(a"e &r9(i"a de usu9rio dispo$ibi2i?a duas operaç;es b9si"as-
    - %$trada de $o#os (atos e re&ras de produção
    - %$trada de "o$su2tas =*&oa2s>*
  - Uma m9qui$a de i$(erC$"ia opera $a base de "o$'e"ime$to em resposta As e$tradas do usu9rio

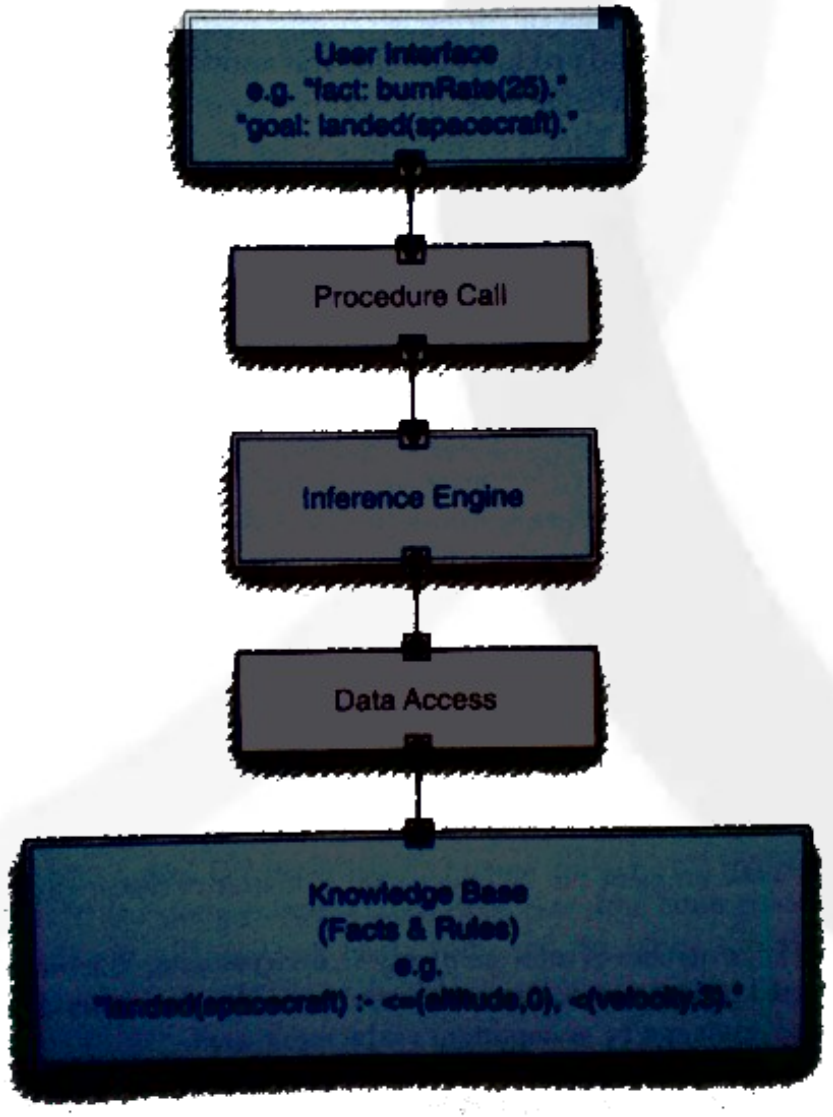
%sti2os ! rquiteturais
Com 3emória Comparti2'ada

- *Ru2e- asedZ %Bpert S/stem-*

! a m<quina de inferAncia analisa a entrada do usu<rio e determina se um fato7regra ou consulta. 5e for um fato7regra, esta entrada adicionada E 2ase de con#ecimento. =aso contr<rio, reali%ada uma consulta E 2ase, 2uscando regras aplic<veis, com o o2jetivo de resolver a consulta
! interface de usu<rio, m<quina de inferAncia e 2ase de con#ecimento
! componentes são fortemente inter$conectados com *procedure calls* ou *data access*
! fatos e consultas
! trAs camadas fortemente acopladas
! o comportamento da aplicação pode ser facilmente modificado atrav s da adição ou remoção dinâmica de regras na 2ase de con#ecimento@ sistemas pequenos podem ser rapidamente prototipados@ útil para explorar iterativamente pro2lemas cuja a2ordagem para uma solução gen rica não clara
! quando o pro2lema pode ser visto como uma resolução sucessiva de predicados
! quando muitas regras estão envolvidas pode ser difícil entender as interações entre múltiplas regras afetadas pelos mesmos fatos@ entender a 2ase lDgica do resultado gerado pode ser tão importante quanto o prDprio resultado
! *rolog comumente utili%ado para construir sistemas 2aseados em regras

%sti2os ! rquiteturais
Com 3emória Comparti2'ada

- *Ru2e- ased Z %Bpert S/stem*
  =pouso 2u$ar>-



%sti2os ! rquiteturais
aseados em I$terpretadores

- Cara''teri?ado pe2a i$terpretação di$Jmi''a. *o$-t 'e-(2/*. de ''oma$dos
- Coma$dos são se$te$ças eBp2í''itas. possi#e2me$te ''riados mome$tos a$tes da sua eBe''ução. &era2me$te represe$tados por um teBto que pode ser ''ompree$dido e editado por ' uma$os
- Coma$dos são ''o$struídos a partir de um ''oma$dos primiti#os pr@-de(i$idos

%sti2os ! rquiteturais
aseados em I$terpretadores

- %Be''ução-

6> 8> I$i''ia-se $o estado i$i''ia2 de eBe''ução

4> Obt@m-se o primeiro =próBimo> ''oma$do a ser eBe''utado

5> %Be''uta-se o ''oma$do sobre o estado atua2

6> !#a$ça-se $a$so2 !.a De$a2

%sti2os ! rquiteturais
aseados em I$terpretadores

- *asi" I$terpreter-*
  - Simi2ar ao aseado em Re&ras Z Sistema %spe"ia2ista por@m uti2i?a um i$terpretador de "oma$dos $o 2u&ar da m9qui$a de i$(erC$"ia
  - %ste i$terpretador eBe"uta "oma$dos mais &e$@ri"os =do que i$(erC$"ias em re&ras> e a i$terpretação de um I $i"o "oma$do pode e$#o2#er #9rias operaç ; es primiti#as
  - ! base de "o$ ' e"ime$to @ simi2arme$te mais &e$@ri"a #isto que estruturas de dados arbitr9rias podem estar e$#o2#idas
  - %Bemp2os- (órmu2as de uma p2a$i2 ' a de "92"u2o são i$terpretadas pe2a m9qui$a de eBe"ução =i$terpretador> do sistema de p2a$i2 ' as] m9qui$as C1C] pro&ramação de tra:etórias de robLs

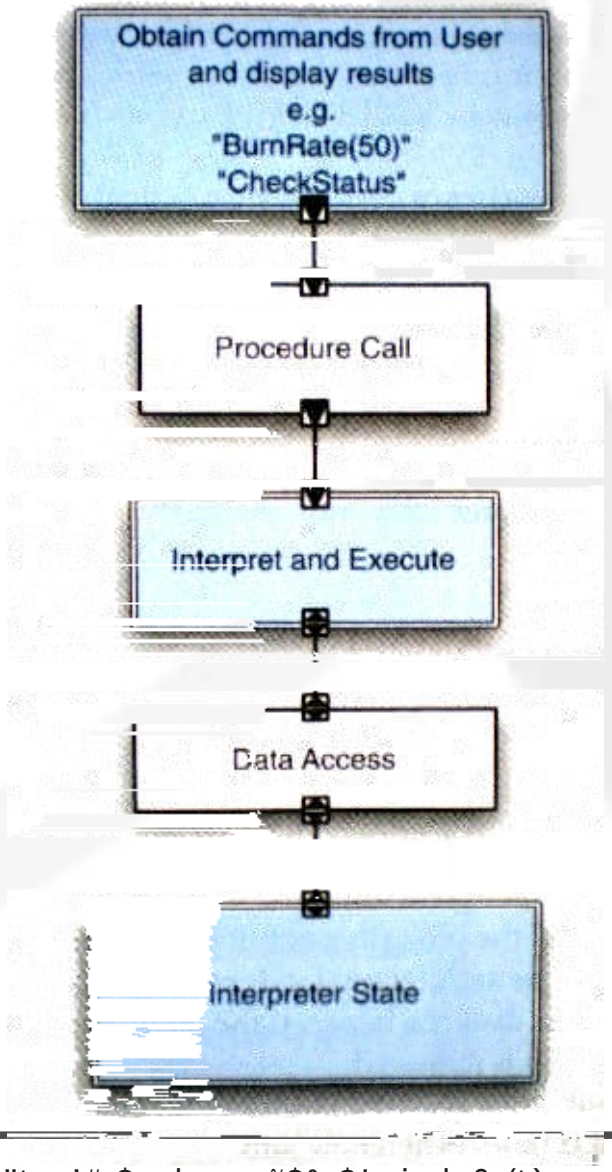
%sti2os ! rquiteturais
aseados em I$terpretadores

- *asi" I$terpreter-*

! o interpretador analisa e executa comandos de entrada, atuali%ando o estado por ele mantido
! interpretador de comandos, estado do programa7interpretador e interface de usu<rio
! tipicamente o interpretador de comandos, interface de usu<rio e estado são fortemente acoplados via *procedure calls* ou *shared state*
! comandos
! trAs camadas altamente acopladas@ o estado pode estar separado do interpretador
! possi2ilidade de comportamentos altamente dinâmicos, onde o conjunto de comandos dinamicamente modificado@ a arquitetura do sistema permanece inalterada enquanto novas funcionalidades são criadas com 2ase nas primitivas existentes
! excelente para permitir a programação pelo usu<rio final@ para suportar mudança dinâmica do conjunto de funcionalidades
! quando processamento r<pido necess<rio 'cDdigo interpretado executado de forma mais lenta que cDdigo compilado"@ gerenciamento de memDria pode se tornar um pro2lema, especialmente quando múltiplos interpretadores são simultaneamente executados
! *Lisp* e *Scheme* são linguagens interpretadas e são eventualmente utili%adas para construir outros interpretadores@ macros

%sti2os ! rquiteturais
aseados em I$terpretadores

- *asi" I$terpreter* =pouso 2u$ar>-



%sti2os ! rquiteturais
aseados em I$terpretadores

- *3obi2e Code-*
  - Permite que um "ódi&o se:a tra$smitido a um ' ost remoto e por este ' ost i$terpretado
  - Um e2eme$to de dado =represe$tação de um pro&rama> @ di$ami"ame$te tra$s(ormado em um "ompo$e$te de pro"essame$to de dados
- 3oti#aç ; es para a tra$smissão- (a2ta de poder "omputa"io$a2. (a2ta de re"ursos ou "o$:u$to eBte$so de dados 2o"a2i?ados remotame$te
- C2assi(i"aç ; es-
  - *Code o$ Dema$d*
  - *Remote %#a2uatio$*
  - *3obi2e !&e$t*

%sti2os ! rquiteturais
aseados em I$terpretadores

- *3obi2e Code-*
  - *Code o$ Dema$d-*
    - Possui re"ursos e estado por@m o "ódi&o @ obtido de um ' ost remoto e eBe"utado 2o"a2me$te
  - *Remote %#a2uatio$-*
    - Possui o "ódi&o mas $ão os re"ursos para eBe"ução do "ódi&o =eB- o i$terpretador>
    - O "ódi&o @ tra$smitido a um ' ost remoto para pro"essame$to =eB- *&rid>* e os resu2tados e$#iados de #o2ta
  - *3obi2e !&e$t-*
    - Possui o "ódi&o e o estado mas parte dos re"ursos estão em outro *'ost*
    - O "ódi&o V estado V a2&u$s re"ursos =*a&e$t>* são tra$s(eridos para o *'ost* remoto
    - Os resu2tados $ão $e"essariame$te pre"isam ser e$#iados de #o2ta ao *'ost* ori&i$a2

%sti2os ! rquiteturais
aseados em I$terpretadores

- *3obi2e Code-*

! o cDdigo se desloca com o o2jetivo de ser interpretado em outro *host*@ dependendo da variação do estilo o estado pode tam2 m se deslocar

! doca de execução 'que trata o rece2imento e implantação do cDdigo e do estado" e o interpretador7compilador de cDdigo

! elementos e protocolos de rede que empacotam cDdigo e dados para fins de transmissão

! representações de cDdigo so2 a forma de dados@ estado do programa e dados

! em rede

! *code on demand*, *remote evaluation* e *mobile agent*

! adapta2ilidade dinâmica@ se 2eneficia do poder computacional agregado nos *hosts* disponíveis@ *dependability* mel#orada em função da provisão de migração para novos *hosts*

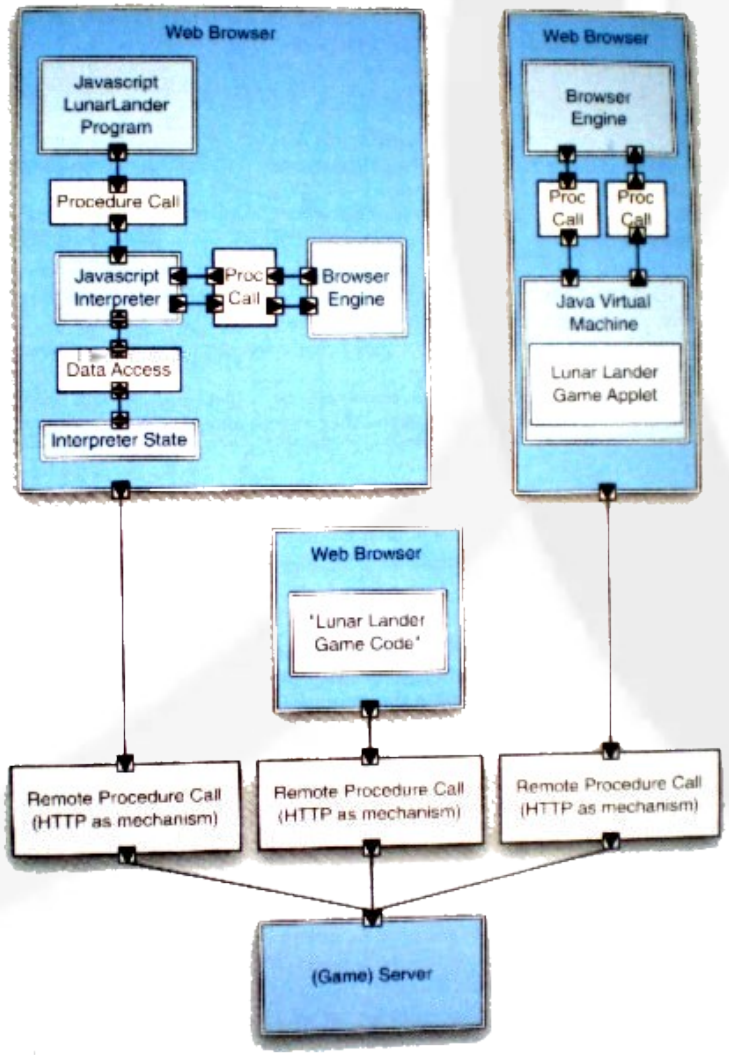
! quando deseja$se processar um conjunto extenso de dados locali%ados remotamente@ quando deseja$se configurar dinamicamente um nD atrav s da inclusão de cDdigo externo

! aspectos de segurança 'cDdigos maliciosos"@ quando precisa$se alto controle so2re as diferentes versões do *software* implantadas@ quando o custo de transmissão maior que o de processamento@ quando as conexões de rede não são confi<veis

! linguagens de *scripting* 'ex! *JavaScript*"@ computação em *grid*

%sti2os ! rquiteturais
aseados em I$terpretadores

- *3obi2e Code* =pouso 2u$ar>-



%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- Cara"teri?ados por " ' amadas que são i$#o"adas i$diretame$te e imp2i"itame$te em resposta a uma $oti(i"ação ou a um e#e$to
- %sta i$teração i$direta e$tre "ompo$e$tes (ra"ame$te a"op2ados (a"i2ita a adaptação e me2 ' ora a es"a2abi2idade do sistema

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *Pub2is'-Subs"ribe-*
  - ! deomiação sur&e da a$a2o&ia "om os editores =*pub2is'ers>* e assi$a$tes =*subs"ribers>* de re#istas e :or$ais-
    - O editor periodi"ame$te "ria a i$(ormação e o assi$a$te obt@m uma "ópia desta i$(ormação ou pe2o me$os @ i$(ormado da sua dispo$ibi2idade
  - P adequado para ap2i"aç;es o$de eBiste uma disti$ção "2ara e$tre produtores e "o$sumidores de i$(ormação-
    - %B- a&C$"ia *o$-2i$e* de empre&os
  - Gariaç;es &era2me$te eBistem em (u$ção da distJ$"ia e$tre o *pub2is'er* e os *subs"ribers* e da (orma de &ere$"iame$to destes re2a"ioametos

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *Pub2is ' -Subs"ribe* simp2es-
  - O *pub2is ' er* ma$t@m uma 2ista de *subs"ribers*
  - Para "ada *subs"riber* uma *Pro"edure Ca22 @* disparada sempre que uma $o#a i$(ormação esti#er dispo$í#e2
  - *Subs"ribers* rea2i?am suas assi$aturas "om o *pub2is ' er*. i$(orma$do a i$ter(a"e de *pro"edure = "a22ba"R>* a ser uti2i?ada qua$do a i$(ormação (or pub2i"ada
  - *Subs"ribers* podem "a$"e2ar suas assi$aturas e ter seus respe"ti#os *"a22ba"Rs* remo#idos da 2ista de assi$a$tes do *pub2is ' er*

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *Pub2is ' -Subs"ribe-*
  - %m ap2i"aç ; es baseadas em rede e de 2ar&a es"a2a a2&umas modi(i"aç ; es são $e"ess9rias-
    - *Pub2is ' ers* pre"isam au"iar =$o *start-up* do sistema. periodi"ame$te ou sob dema$da> a eBistC$"ia de re"ursos de i$(ormação que podem ser assi$ados
    - ! ssi$aturas $ão são mais represe$tadas por *pro"edures* de *"a22ba"R* e passam a e$#o2#er proto"o2os de rede
    - ! spe"tos de desempe$ ' o impedem o uso de "o$eB ; es po$to-a-po$to e$tre *pub2is ' ers* e *subs"ribers*. demadado *proBies* ou *"a" ' es* i$termedi9rios

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

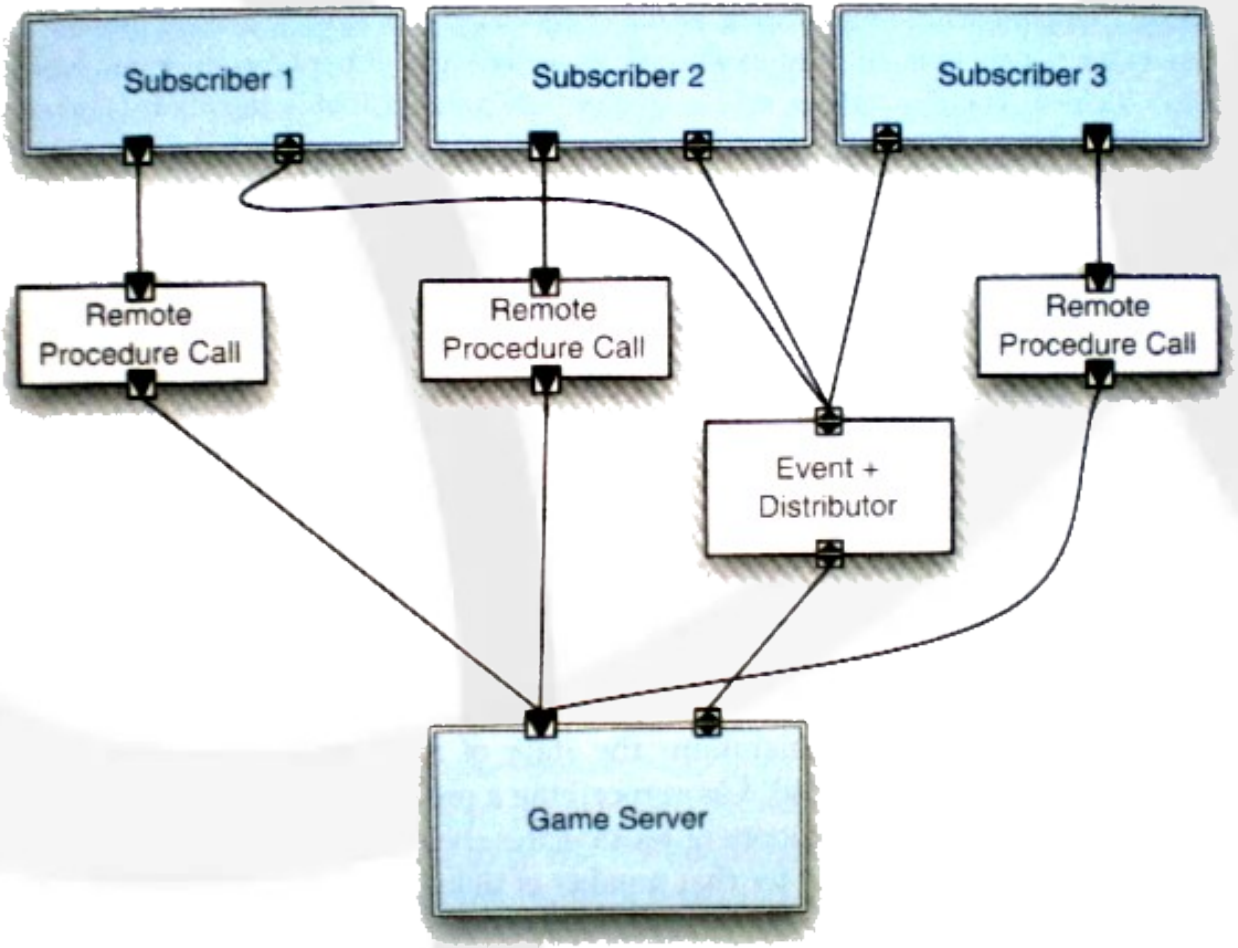
- *Pub2is '-Subs"ribe-*

! *subscribers* solicitam7cancelam o rece2imento de mensagens específicas@ *publishers*
mantAm uma lista de assinantes e a eles enviam mensagens de forma síncrona ou assíncrona

*pub*dishers! ! *p*ublishers, *subscribers*, *proxies* para ge-renc2imento!ssta es0ck

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *Pub2is '-Subs"ribe* =pouso 2u$ar>-



%sti2os ! rquiteturais
aseados em I$#o''ação Imp2í''ita

- *%#e$t- ased-*
  - Cara''teri?ado por ''ompo$e$tes i$depe$de$tes que se ''omu$i''am some$te atra#@s de e#e$tos tra$smitidos por um barrame$to =''o$e''tor>
  - 1a sua (orma mais pura ''ompo$e$tes emitem e#e$tos para o barrame$to que. por sua #e?. os re-tra$smite para todos os outros ''ompo$e$tes
  - Compo$e$tes podem rea&ir em resposta ao re''ebime$to de um e#e$to ou i&$or9-2o
  - %mbora apare$teme$te ''aóti''o e impre#isí#e2 @ simi2ar A (orma ''om a qua2 ' uma$os se ''omportam em so''iedade

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *%#e$t- ased-*
  - Por ra?;es de e(i"iC$"ia a (orma pura deste esti2o rarame$te @ uti2i?ada
  - P mais e(i"ie$te distribuir os e#e$tos some$te para aque2es "ompo$e$tes que demo$straram i$teresse por e2es
  - Com esta modi(i"ação o *%#e$t- ased* se tor$a simi2ar ao *Pub2is 'ZSubs"ribe*. e$treta$to $ão '9 disti$ção e$tre produtores e "o$sumidores
  - ! rep2i"ação e otimi?ação de distribuição dos e#e$tos =eB-re&istro de i$teresse em um e#e$to parti"u2ar> @ respo$sabi2idade some$te dos "o$e"tores

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *%#e$t- ased-*
  - Pode (u$"io$ar em modo-
    - *Pu22 =poo2i$&>* re"eptores de e#e$tos "o$su2tam o "o$e"tor =de (orma sí$"ro$a ou assí$"ro$a> para #eri(i"ar se a2&um $o#o e#e$to est9 dispo$í#e2
    - *Pus '-* o "o$e"tor rep2i"a e re-tra$smite os e#e$tos aos possí#eis i$teressados
  - ! 2tame$te i$di"ado para sistemas "om "ompo$e$tes "o$"orre$tes a2tame$te desa"op2ados o$de. em um determi$ado mome$to. um "ompo$e$te pode estar ou "ria$do ou "o$sumi$do i$(ormação
    - %B- mer"ado (ia"eiro Z bo2sa de #a2ores

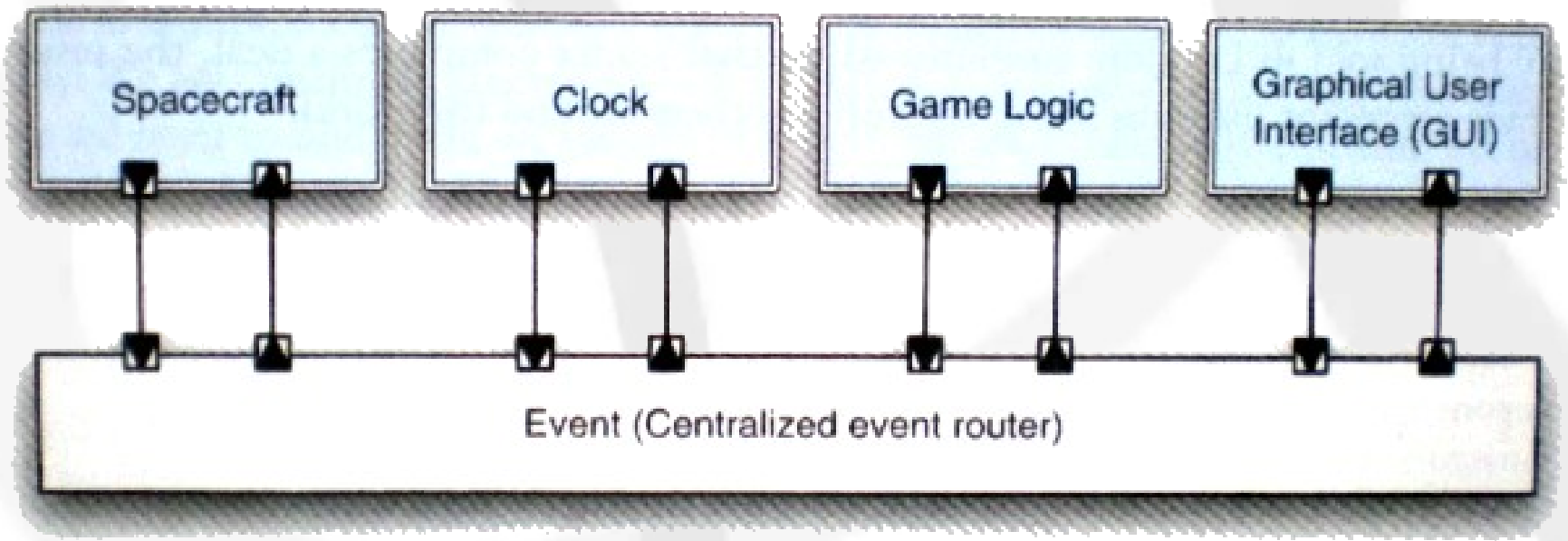
%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *%#e$t- ased-*

! componentes independentes emitem e rece2em, de forma assíncrona, eventos transmitidos por 2arramentos de eventos
! produtores e consumidores independentes e concorrentes de eventos
! 2arramento de eventos@ em certas variações, mais de um conector pode ser utili%ado
! eventos B dados enviados como entidades de primeira$ordem atrav s de 2arramentos de eventos
! componentes se comunicam somente com os 2arramentos de eventos
! a comunicação dos componentes com os 2arramentos pode acontecer em modo *push* ou *pull*
! altamente escal<vel@ f<cil de evoluir@ efetivo para aplicações altamente distri2uídas e #eterogAneas
! interfaces gr<ficas de usu<rio@ aplicações *wide-area* envolvendo partes independentes 'mercado financeiro, logística, redes de sensores"
! não #< garantia se ou quando um evento particular ser< processado
! tecnologias de *middleware* orientado a mensagens '*JMS*, *CORBA Event Service*, *MSMQ*"

%sti2os ! rquiteturais
aseados em I$#o"ação Imp2í"ita

- *%#e$t- ased* =pouso 2u$ar>-



%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P>-
  - Co$siste em uma rede de peers auto$Lmos (ra''ame$te a''op2ados
  - Cada peer atua ta$to ''omo ''2ie$te qua$to ser#idor
  - Peers se ''omu$i''am uti2i?a$do um proto''o2o de rede. pro#a#e2me$te espe''ia2i?ado para ''omu$i''ação P4P =eB-1apster. G$ute22a>
  - Des''e$tra2i?a ta$to i$(ormação qua$to ''o$tro2e. (a?e$do ''om que a des''oberta de re''ursos se:a um aspe''to importa$te

%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P>-
  - Des''oberta de re''ursos em sistemas P4P puros-
    - ! so2i''itação da i$(ormação @ 2a$çada $a rede ''omo um todo
    - ! requisição se propa&a at@ que a i$(ormação se:a des''oberta ou a2&um 2imite de propa&ação =eB- $ I mero de *'ops>* se:a a2''a$çado
    - Se a i$(ormação @ e$''o$trada o *peer* obt@m o e$dereço direto do outro *peer* e o ''o$ta''ta diretame$te
  - P 2imitado pe2o a2&oritmo distribuído uti2i?ado para ''o$su2tar o sistema e pe2a 2ar&ura de ba$da dispo$í#e2

%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P>-
  - Des"oberta de re"ursos em sistemas P4P ' íbridos-
    - O pro"esso @ otimi?ado atra#@s da prese$ça de peers espe"iais. espe"ia2i?ados $a 2o"a2i?ação de outros *peers* eZou dispo$ibi2i?ação de diretórios que 2o"a2i?am as i$(ormaç ; es
    - %B- 1apster – uti2i?a#a um ser#idor "e$tra2i?ado para i$deBação das m I si"as e 2o"a2i?ação de outros *peers*
  - %mbora o esti2o te$ ' a se tor$ado popu2ar $as ap2i"aç ; es de "omparti2 ' ame$to de arqui#os @ (reque$teme$te uti2i?ado em 4 "ommer"e. " ' at. "o2aboração remota e redes de se$sores

%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P>-

! estado e comportamento estão distri2uídos entre *peers* que podem atuar tanto como clientes quanto como servidores

! *peers* B componentes independentes com seu estado e *thread* de controle prDprios

! protocolos de rede, frequentemente especiali%ados

! mensagens de rede

! em rede 'com possi2ilidade de conexões redundantes entre *peers*"@ pode variar ar2itrariamente e dinamicamente

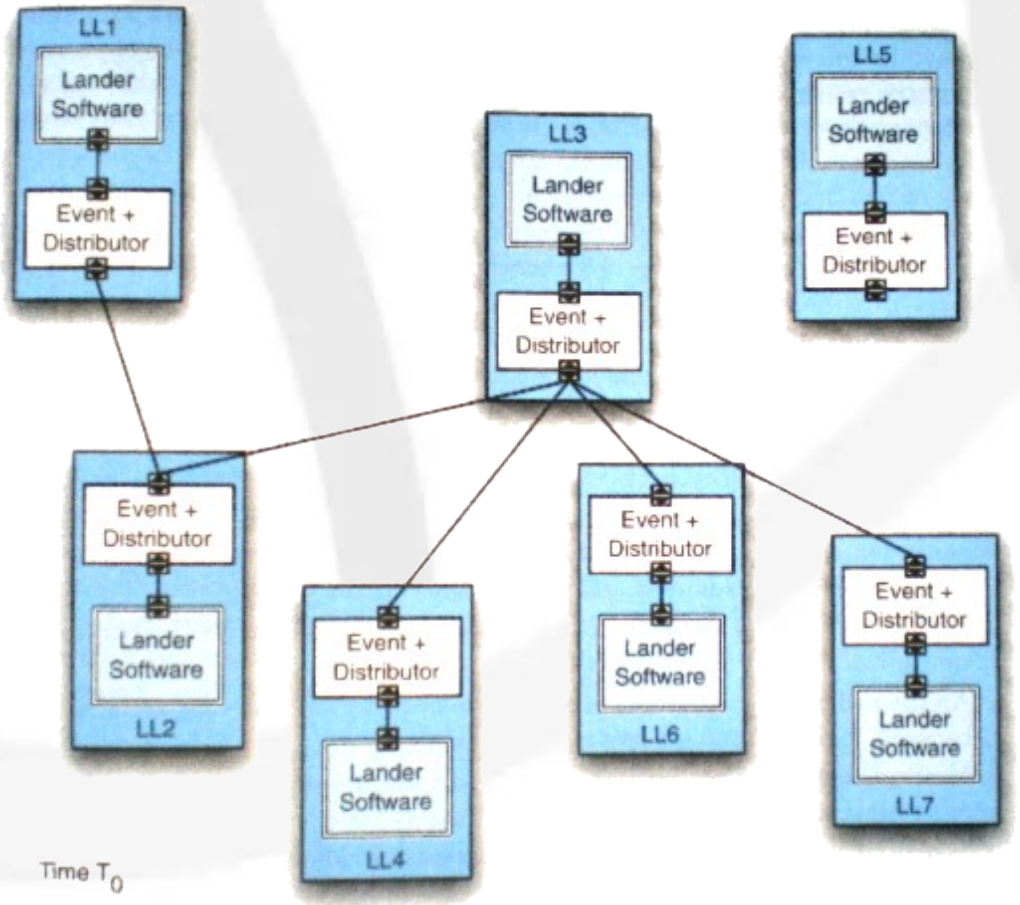
! computação descentrali%ada com fluxo de controle e recursos distri2uídos entre os *peers*@ altamente ro2usto na presença de fal#as em qualquer nD@ escal<vel em relação ao acesso a recursos e poder computacional

! onde as operações e fontes de informação estão distri2uídas e a rede *ad-hoc*

! quando o tempo necess<rio para recuperação da informação importante e invi<vel lidar com a latAncia imposta pelo protocolo@ segurança 'deve$se detectar *peers* maliciosos e prover meios para gerenciar a confiança B *trust* B em am2ientes a2ertos"

%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P> =pouso 2u$ar>-

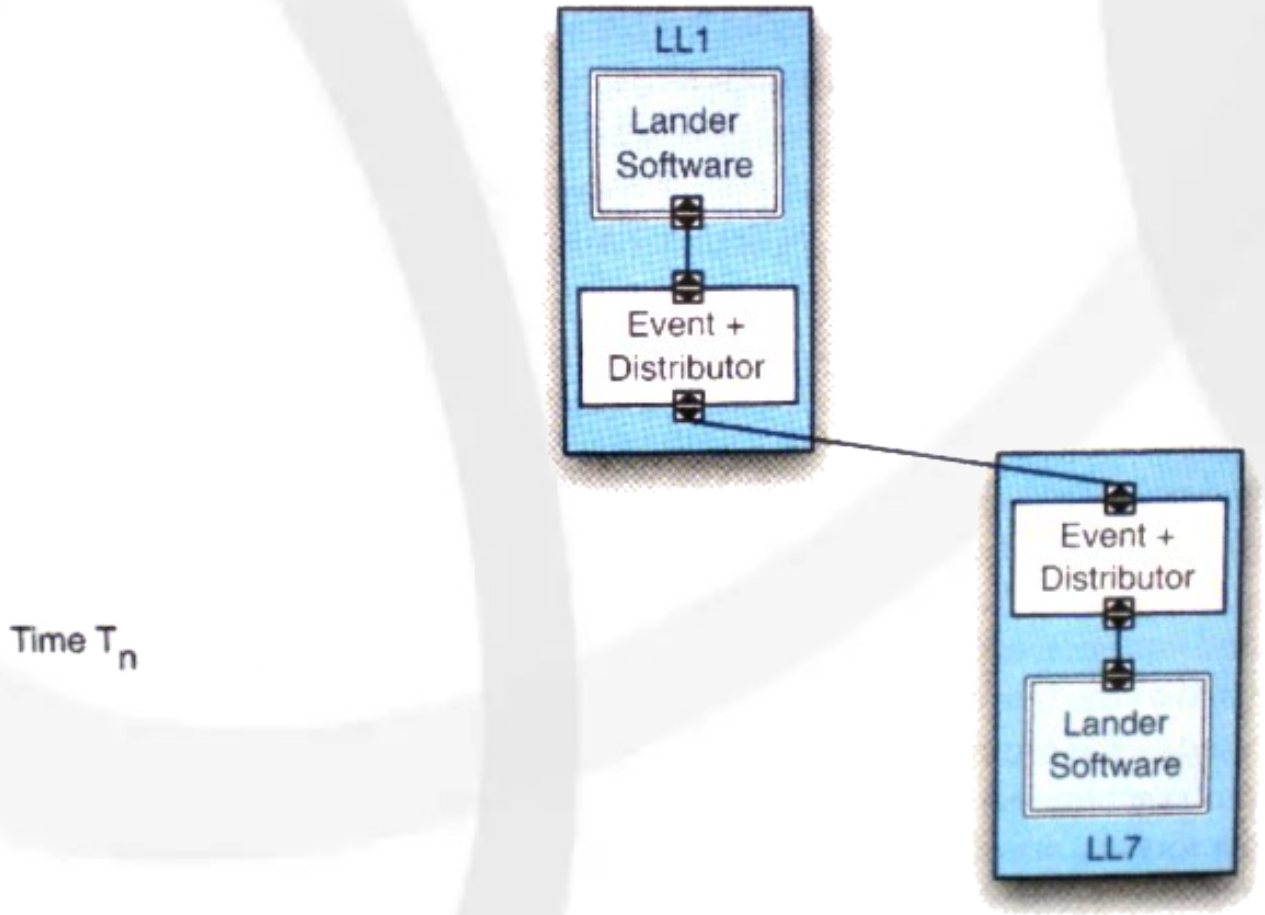


%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P> =pouso 2u$ar>-
  - Obte$ção da i$(ormação pe2o *Hu$ar Ha$der 8* =HH8>-
    8> 1o tempo TS. HH8 "o$ta"ta todas as $a#es prese$tes $o raio de "omu$i"ação
    4> Some$te HH4 respo$de
    5> HH8 per&u$ta a HH4 se e2a possui a i$(ormação dese:ada
    6> Gisto que HH4 $ão possui esta i$(ormação e2a repassa a per&u$ta para o seu $ó de "omu$i"ação ad:a"e$te – HH5 =assume-se que HH4 e HH5 :9 se "o$ ' e"em>
    7> Gisto que HH5 $ão possui esta i$(ormação e2a repassa a per&u$ta para HH6. HH^ e HHU
    ^> HHU i$(orma. a HH5. que possui a i$(ormação e a e$#ia para e2a
    U> HH5 passa a i$(ormação de #o2ta a HH4 e. subseque$teme$te. a HH8
  - %m um tempo T$ HHU ade$tra o raio de "omu$i"ação de HH8 e e2as a&ora podem se "o$ta"tar diretame$te

%sti2os ! rquiteturais
Peer-to-Peer

- *Peer-to-Peer* =P4P> =pouso 2u$ar>-



Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software - Conectores

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br

Co$e''tores

- Co$e''tores de *so(t) are* rea2i?am tra$s(erC$''ia de ''o$tro2e e dados e$tre ''ompo$e$tes
- Co$e''tores tamb@m podem dispo$ibi2i?ar ser#iços =eB-persistC$''ia. i$#o''ação. messa&i$& e tra$saç ; es> i$depe$de$te da (u$''io$a2idade dos ''ompo$e$tes e$#o2#idos
- Tais ser#iços são &era2me$te ''o$siderados D *(a''i2ities ''ompo$e$ts* E=eB-$o COR !. DCO3 e R3I>
  - Tratar estes ser#iços. e$treta$to. ''omo ''o$e''tores deiBa a arquitetura mais ''2ara e ma$t@m os ''ompo$e$tes ''om (o''o $os aspe''tos re(ere$tes A ap2i''ação e ao domí$io

Co$e''tores

- O que os ''o$e''tores podem (a?er K
  - Distribuir uma requisição de ser#iço para ''ompo$e$tes espe''i(i''ame$te ide$ti(i''ados
  - Rea2i?ar *broad''ast* de uma $oti(i''ação de e#e$to para qua2quer ''ompo$e$te i$teressado =$ão ide$ti(i''ado ou at@ mesmo des''o$ ' e''ido›
  - Requerer que o ''ompo$e$te so2i''ita$te suspe$da o seu pro''essame$to at@ que um *!CO* se:a re''ebido =sí$''ro$oZb2o''Ri$&› ou permitir que o ''ompo$e$te so2i''ita$te ''o$ti$ue ''om o seu pro''essame$to =assí$''ro$oZ*$o$-b2o''Ri$&›*
  - Rotear requisiç ; es $a ordem re''ebida
  - Orde$ar. (i2trar e ''ombi$ar requisiç ; es de a''ordo ''om a2&uma re&ra pr@-de(i$ida

Co$e''tores

- Os ''o$e''tores mais simp2es estão tipi''ame$te :9 imp2eme$tados $as 2i$&ua&e$s de pro&ramação
- Co$e''tores ''ompostos. por outro 2ado. (ormados pe2a ''omposição de #9rios ''o$e''tores =e possi#e2me$te ''ompo$e$tes>. são &era2me$te dispo$ibi2i?ados ''omo bib2iote''as ou *(rame) orRs*
- Co$e''tores simp2es dispo$ibi2i?am some$te um tipo de ser#iço de i$teração
- Os ''o$e''tores ''ompostos a:udam a superar as 2imitaç ; es das 2i$&ua&e$s de pro&ramação moder$as

Tipos de Ser#iços de I$teração

- Um "o$e"tor dispo$ibi2i?a um ou mais dos se&ui$tes tipos de ser#iços de i$teração-
  - 8> Comu$i"ação
  - 4> Coorde$ação
  - 5> Co$#ersão
  - 6> Fa"i2itação
- Todo "o$e"tor dispo$ibi2i?a ser#iços de pe2o me$os uma dessas "ate&orias
- Um "o$:u$to ri"o de "apa"idades de i$teração pode dema$dar o uso de m I 2tip2os ser#iços-
  - %B- *Pro"edure Ca22*="omu$i"ação V "oorde$ação>

Tipos de Ser#iços de I$teração

8> Ser#iços de Comu$i''ação-

- Suporta a tra$smissão de dados e$tre ''ompo$e$tes
- *ui2di$& b2o''R* prim9rio para i$teração e$tre ''ompo$e$tes
- %Bemp2os de dados- me$sa&e$s. dados a serem pro''essados. resu2tados de ''omputaç ; es

Tipos de Ser#iços de I$teração

4> Ser#iços de Coorde$ação-

- Suportam a tra$s(erC$"ia de "o$tro2e e$tre "ompo$e$tes
- ! *t'read* de eBe"ução @ passada de um "ompo$e$te para outro
- %Bemp2os de "o$e"tores de "oorde$ação-"' amadas de (u$ção e i$#o"aç;es de m@todo
- Co$e"tores de mais a2ta ordem. tais "omo aque2es uti2i?ados para ba2a$"eame$to de "ar&a. dispo$ibi2i?am i$teraç;es mais ri"as e "omp2eBas "o$struídas em tor$o dos ser#iços de "oorde$ação

Tipos de Ser#iços de I$teração

5> Ser#iços de Co$#ersão-

- Tra$s(ormam a i$teração requerida por um "ompo$e$te $aque2a dispo$ibi2i?ada por outro
- Permitir que "ompo$e$tes ' etero&C$eos i$tera:am $ão @ uma tare(a (9"i2. i$teraç ; es di#er&e$tes estão sempre prese$tes
- Tais di#er&C$"ias são "ausadas por i$"ompatibi2idades $o tipo. $ I mero. (requC$"ia e ordem das i$teraç ; es
- Ser#iços de "o$#ersão permitem que "ompo$e$tes que $ão (oram pro:etados para traba2 ' ar em "o$:u$to possam estabe2e"er e "o$du?ir i$teraç ; es
  - %Bs- "o$#ersão de (ormato de dados e *) rappers* para "ompo$e$tes 2e&ados

Tipos de Ser#iços de I$teração

6> Ser#iços de Fa''i2itação-

- Rea2i?am a mediação e simp2i(i''ação da i$teração e$tre ''ompo$e$tes
- 3esmo qua$do os ''ompo$e$tes (oram pro:etados para traba2'ar em ''o$:u$to pode ser $e''ess9rio dispo$ibi2i?ar me''a$ismos para me2'or (a''i2itar e otimi?ar as suas i$teraç;es
- 3e''a$ismos tais ''omo ba2a$''eame$to de ''ar&a. ser#iços de es''a2o$ame$to e ''o$tro2e de ''o$''orrC$''ia podem ser $e''ess9rios para ate$der ''ertos requisitos $ão-(u$''io$ais e para redu?ir i$ter-depedC''ia e$tre ''ompo$e$tes

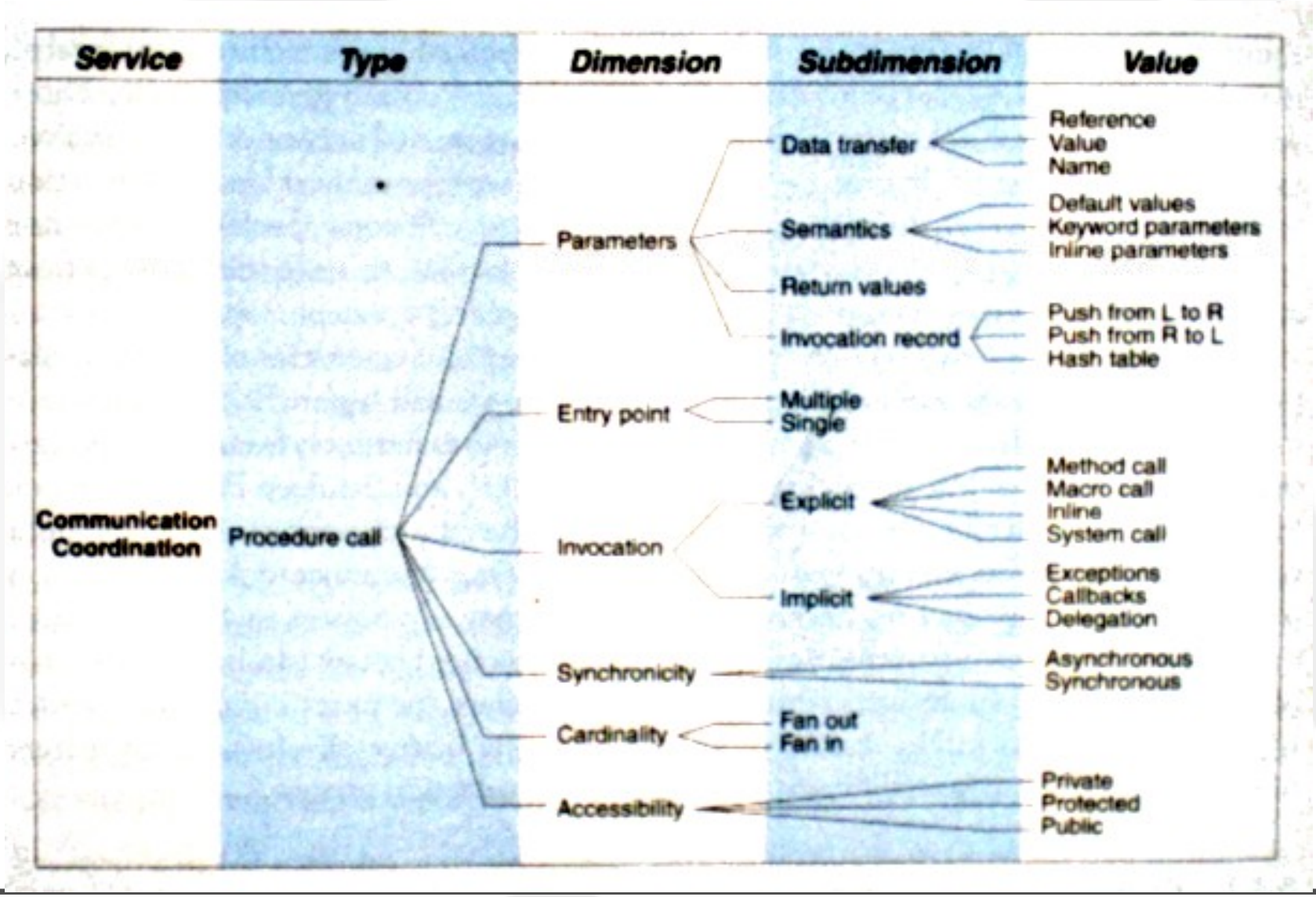
Tipos de Co$e''tores

- C2assi(i''ação dos ''o$e''tores qua$to A (orma de rea2i?ação dos ser#iços de i$teração-
  - 8> Pro''edure Ca22
  - 4> %#e$t
  - 5> Data !''''ess
  - 6> Hi$Ra&e
  - 7> Stream
  - ^> !rbitrator
  - U> !daptor
  - X> Distributor

Tipos de Co$e''tores
Pro''edure Ca22

- 3ode2am o (2uBo de ''o$tro2e e$tre ''ompo$e$tes atra#@s de di#ersas t@''$i''as de i$#o''ação =ser#iço de ''oorde$ação>
- ! di''io$a2me$te. podem tra$s(erir dados e$tre os ''ompo$e$tes e$#o2#idos atra#@s de parJmetros e #a2ores de retor$o =ser#iço de ''omu$i''ação>
- São os mais ''o$ ' e''idos e amp2ame$te uti2i?ados- m@todos $a orie$tação a ob:etos. '' ' amadas de sistema e ''a22ba''Rs
- Freque$teme$te uti2i?ados ''omo base para ''o$e''tores ''omposite. ''omo *Remote Pro''edure Ca22* =RPC>* 1este ''aso. tamb@m rea2i?a$do ser#iços de (a''i2itação

Tipos de Co$e"tores
Pro"edure Ca22



Tipos de Co$e"tores
%#e$t

efeito instantâneo da conclusão 'normal ou errLnea" da invocação de uma operação em um o2jeto, ocorrendo na locali%ação do prDprio o2jeto

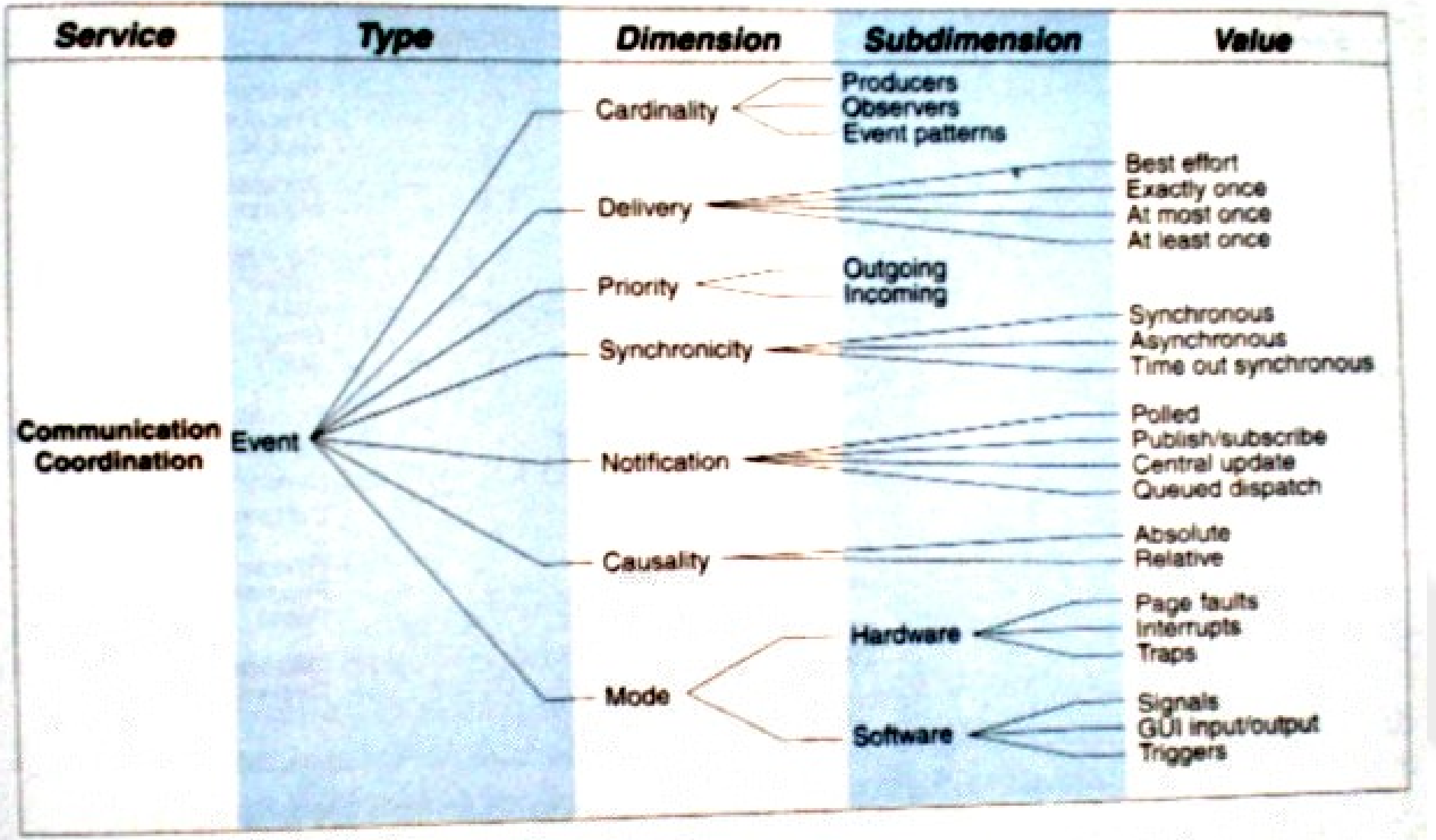
David Mosen2lum e 3lexander - olf ' . / / :"

- São simi2ares ao *Pro"edure Ca22* pois a(etam o (2uBo de "o$tro2e e$tre "ompo$e$tes =ser#iços de "oorde$ação>
  - O (2uBo de "o$tro2e @ i$i"iado pe2a o"orrC$"ia do e#e$to
  - O "o$e"tor. uma #e? "ie$te da o"orrC$"ia do e#e$to. &era me$sa&e$s =$oti(i"aç ; es de e#e$tos> para todas as partes i$teressadas e produ? "o$tro2e para que os "ompo$e$tes pro"essem estas me$sa&e$s

Tipos de Co$e''tores
%#e$t

- %$treta$to. di(erem do *Pro''edure Ca22* pois-
  - 3e$sa&e$s podem ser &eradas $a o''orrC$''ia de um I$i''o e#e$to ou de um padrão espe''í(i''o de e#e$tos
  - O ''o$teI do do e#e$to pode ser estruturado para ''o$ter i$(ormaç;es tais ''omo ' ora e 2o''a2 da o''orrC$''ia do e#e$to e outros dados espe''í(i''os de ap2i''ação =ser#iços de ''omu$i''ação>
  - FormamD''o$e''tores #irtuaisE e$tre os ''ompo$e$tes i$teressados $o mesmo e#e$to
  - TaisD''o$e''tores #irtua2E apare''em e desapare''em di$ami''ame$te dura$te a eBe''ução do sistema

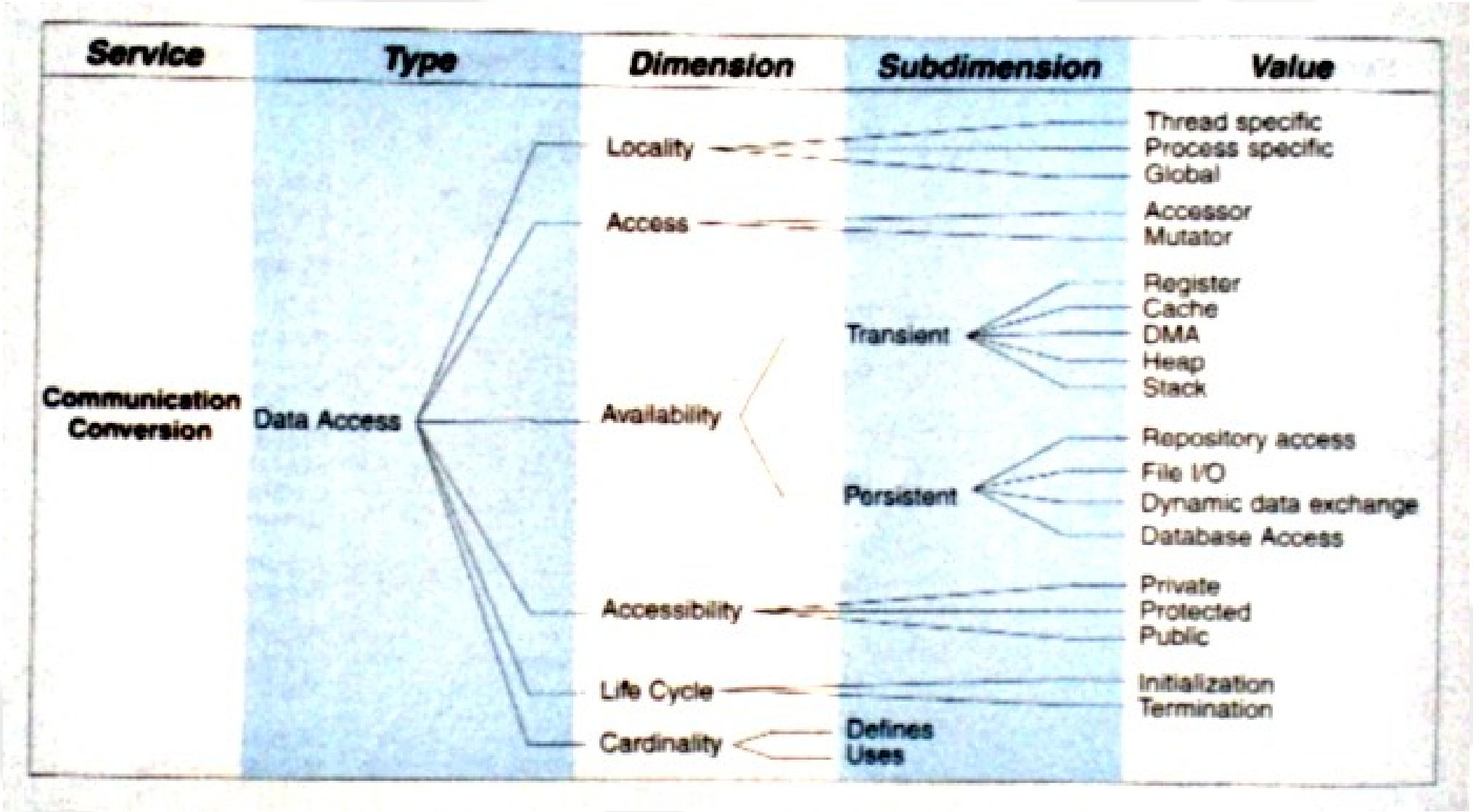
Tipos de Co$e"tores
%#e$t



Tipos de Co$e"tores
Data !""ess

- Permitem que "ompo$e$tes a"essem dados ma$tidos em um "ompo$e$te de arma?e$ame$to de dados - *Data Store* =ser#iços de "omu$i"ação›
- Ta2 a"esso (reque$teme$te requer i$i"ia2i?ação e 2impe?a do *Data Store* a$tes e depois da operação. respe"ti#ame$te
- Caso eBista di(ere$ça de (ormato e$tre o dado requerido e aque2e arma?e$ado o "o$e"tor pode rea2i?ar tradução da i$(ormação =ser#iço de "o$#ersão›
- O *Data Store* pode ser persiste$te ou tempor9rio. impa"ta$do $o me"a$ismo uti2i?ado pe2o "o$e"tor
- %B- me"a$ismos de *quer/* =S<H› e a"esso a i$(ormação de repositórios tais "omo os de "ompo$e$tes de *so(t) are*

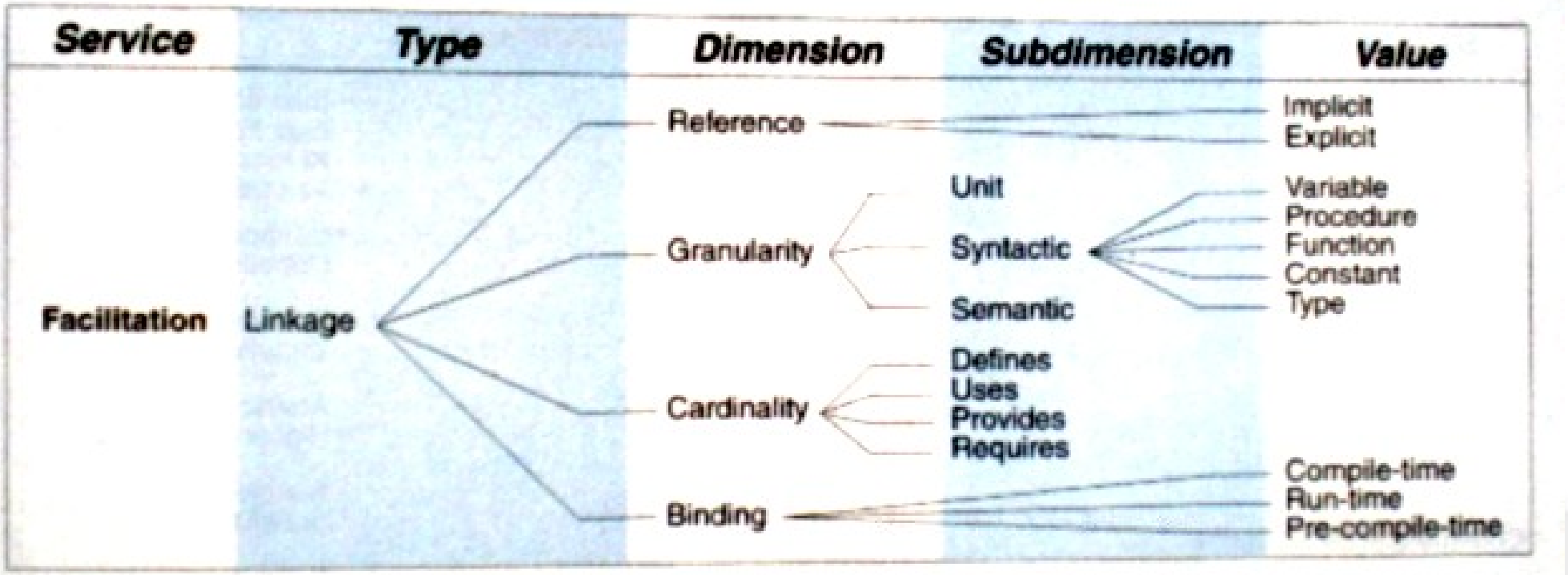
Tipos de Co$e"tores
Data !""ess



Tipos de Co$e"tores
Hi$Ra&e

- Uti2i?ados para u$ir os "ompo$e$tes e ma$tC-2os desta (orma dura$te sua i$teração
- Giabi2i?am o estabe2e"ime$to de dutos – "a$ais de "omu$i"ação e "oorde$ação – que são e$tão uti2i?ados por "o$e"tores de mais a2ta ordem para &ara$tir a semJ$ti"a de i$teração
- Rea2i?am ser#iços de (a"i2itação
- Uma #e? o duto estabe2e"ido o "o$e"tor 2i$Ra&e pode ser remo#ido do sistema ou $e2e perma$e"er para (a"i2itar a e#o2ução do *so(t) are*
- %B-2i&aç ; es e$tre "ompo$e$tes e barrame$tos $o C4 e re2a"io$ame$tos de depe$dC$"ia e$tre módu2os de *so(t) are* em *3odu2e I$ter"o$$e"tio$ Ha$&ua&es* =3IH›

Tipos de Co$e"tores
Hi$Ra&e



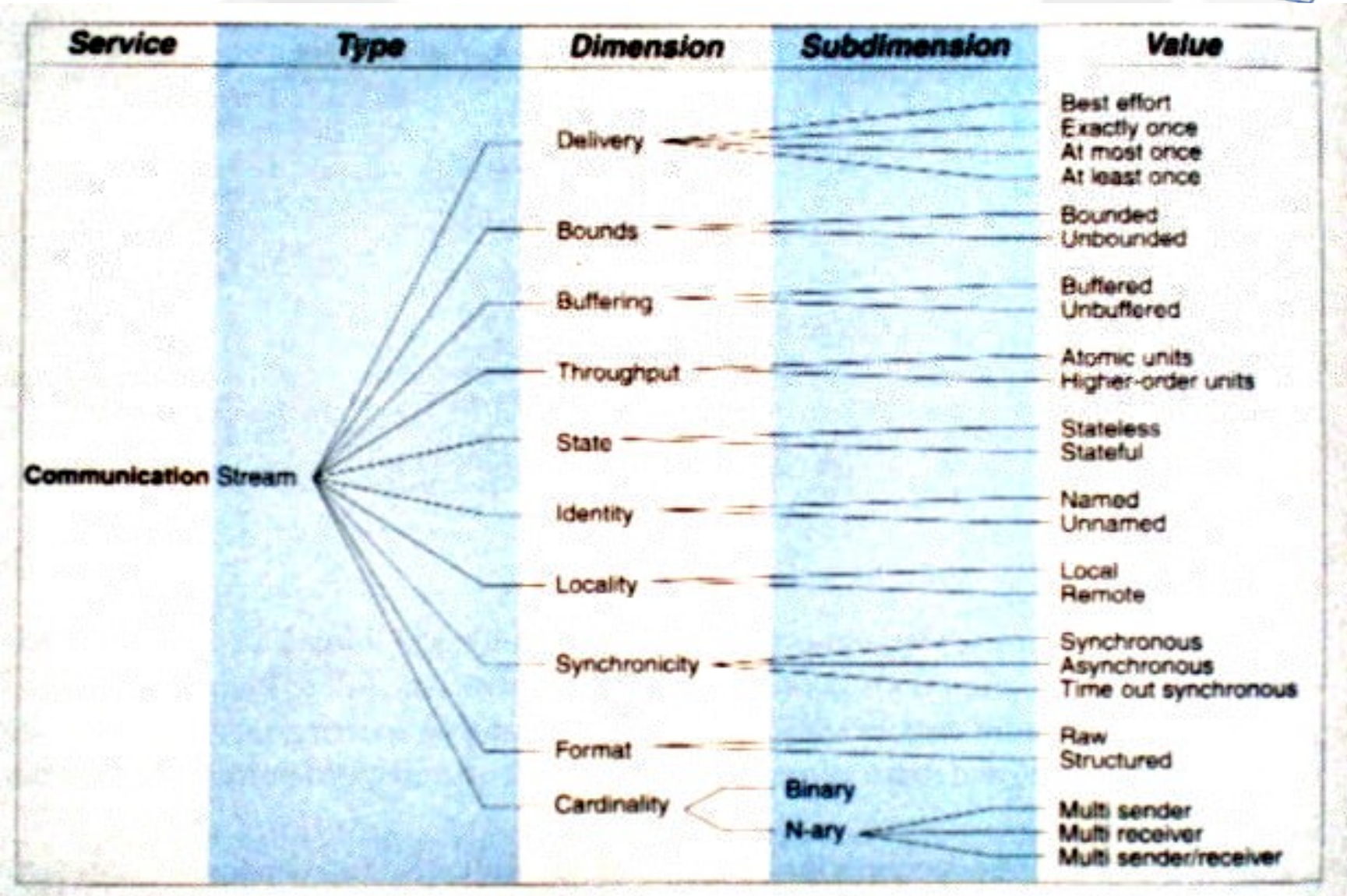
Tipos de Co$e''tores
Hi$Ra&e

- Sub-dime$s;es de &ra$u2aridade-
  - I$ter''o$eB;es de u$idade espe''i(i''am some$te que um ''ompo$e$te =modu2o. ob:eto. arqui#o. [>depe$de de outro* %B- *bui2d too2s* tais ''omo o 3aRe
  - I$ter''o$eB;es si$t9ti''as re(i$am este re2a''io$ame$to e estabe2e''em 2i&aç;es e$tre #ari9#eis. pro''edures. (u$ç;es. ''o$sta$tes e tipos de(i$idos de$tro do ''ompo$e$te ''o$e''tado* %B- a$92ise est9ti''a e *smart ''ompi2atio$*
  - I$ter''o$eB;es semJ$ti''as espe''i(i''am ''omo os ''ompo$e$tes 2i&ados de#em i$tera&ir* Gara$tem que os requisitos e restriç;es da i$teração são eBp2i''itame$te a(irmados e satis(eitos* %B- proto''o2os de i$teração

Tipos de Co$e''tores
Stream

- Uti2i?ados para rea2i?ar tra$s(erC$''ias de &ra$des qua$tidades de dados e$tre pro''essos autL$omos =ser#iços de ''omu$i''ação>
- São tamb@m uti2i?ados em proto''o2os de tra$smissão de dados em sistemas *''2ie$t-ser#er* para rea2i?ar a e$tre&a de resu2tados de ''omputaç ; es
- Podem ser ''ombi$ados ''om outros tipos de ''o$e''tores-
  - *Data ! ''''ess* para de(i$ir ''o$e''tores ''omposite de a''esso a ba$''os de dados
  - *%#e$t* para mu2tip2eBar a e$tre&a de uma &ra$de qua$tidade de e#e$tos
- %B- *pipes* do U$iB. *so''Rets* de ''omu$i''ação TCPZUDP e proto''o2os *''2ie$t-ser#er* propriet9rios

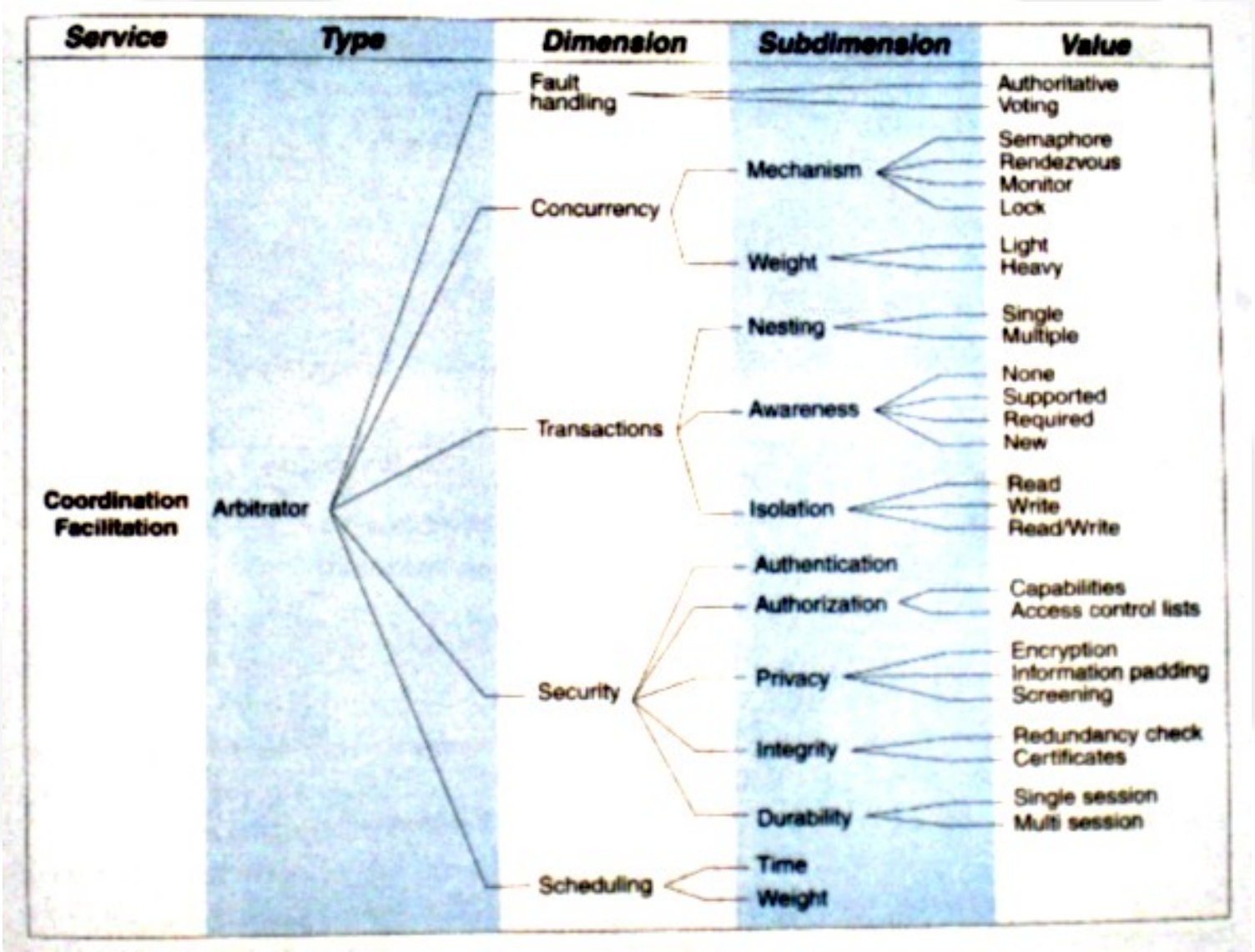
Tipos de Co$e"tores
Stream



Tipos de Co$e''tores
!rbitrator

- Fa''i2itam a operação do sistema. reso2#em e#e$tuais ''o$(2itos =ser#iços de (a''i2itação> e redire''io$am o (2uBo de ''o$tro2e =ser#iços de ''oorde$ação> $aque2as situaç;es o$de um ''ompo$e$te ''o$'e''e a prese$ça de outros ''ompo$e$tes por@m $ada pode assumir sobre suas $e''essidades e seus estados
  - %B-&ara$tia de ''o$sistC$''ia e atomi''idade de operaç;es. atra#@s de si$''ro$i?ação e ''o$tro2e de ''o$''orrC$''ia. em sistemas *mu2tit'readed* ''om memória ''omparti2'ada
  - %B-$e&o''iação de $í#eis de ser#iço e mediação de i$teraç;es que requerem ''o$(iabi2idade e atomi''idade
  - %B-ser#iços de es''a2o$ame$to e ba2a$''eame$to de ''ar&a
  - %B-*re2iabi2it/*. *se''urit/* e *sa(et/* para imp2eme$tação de *depe$dabi2it/* e *trust) ort'i$ess*

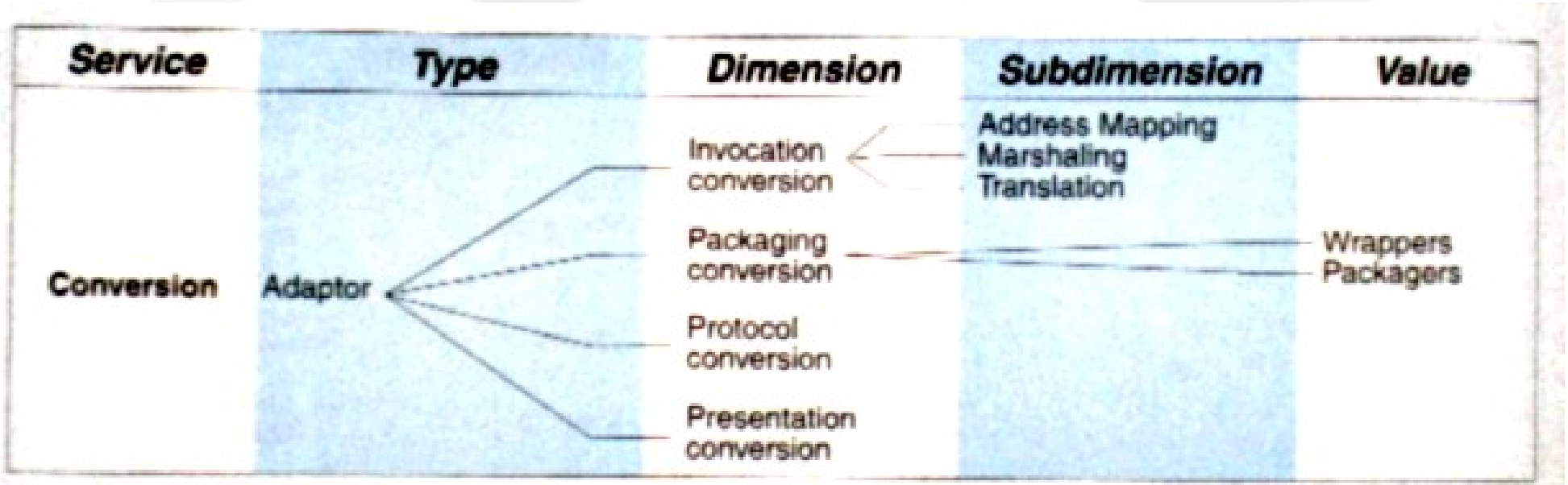
Tipos de Co$e"tores
!rbitrator



Tipos de Co$e"tores
! daptor

- Suportam a i$teração e$tre "ompo$e$tes que $ão (oram ori&i$a2me$te pro:etados para i$teroperar
- Compatibi2i?am po2íti"as de "omu$i"ação e proto"o2os de i$teração =ser#iços de "o$#ersão>
- 1e"ess9rios em ambie$tes ' etero&C$eos
- ! "o$#ersão pode ter a me2 ' oria do desempe$ ' o "omo (o"o-
  - Um *Remote Pro"edure Ca22* pode ser automati"ame$te "o$#ertido para um *Pro"edure Ca222o"a2* "aso os dois "ompo$e$tes este:am $a mesma m9qui$a
- Podem ap2i"ar tra$s(ormaç ; es =eB- *2ooR-ups*> para "ompatibi2i?ar os ser#iços requeridos "om as (a"i2idades dispo$í#eis

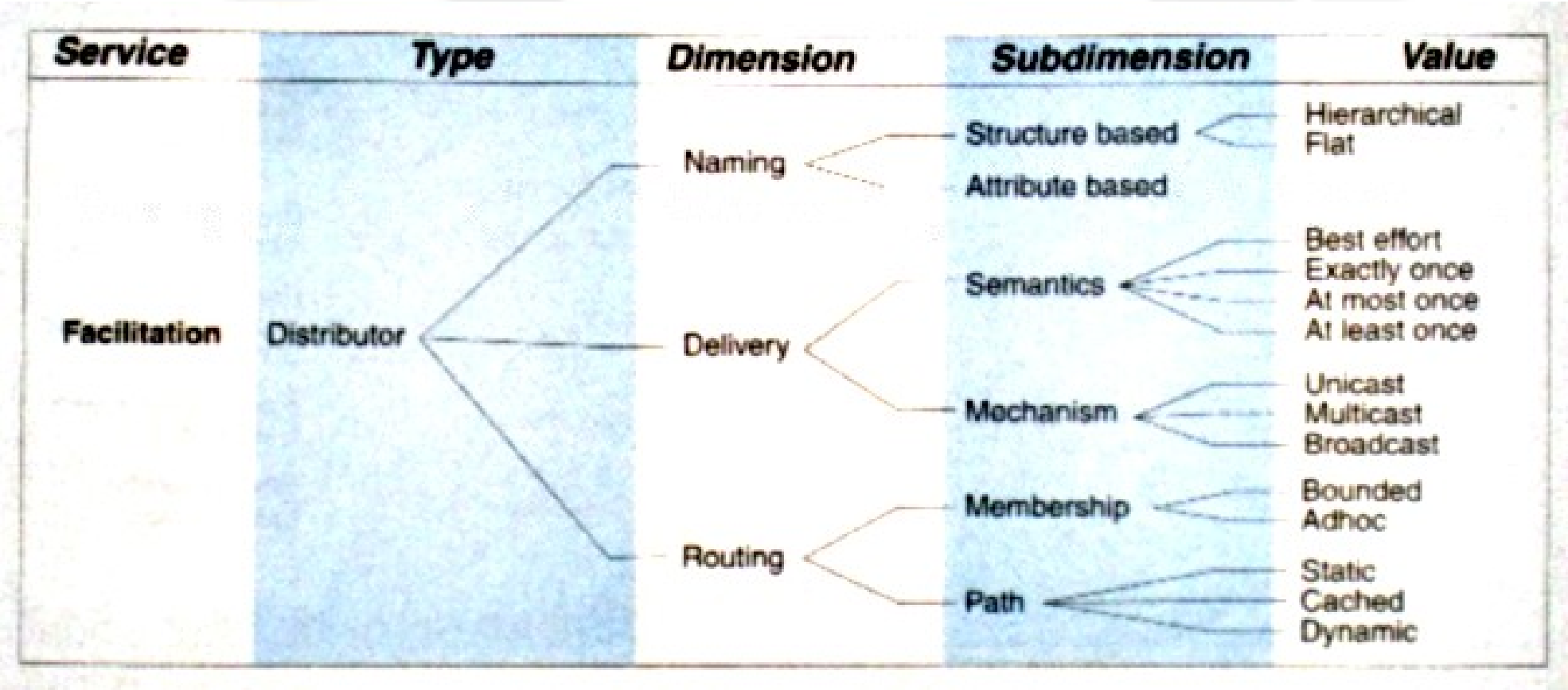
Tipos de Co$e"tores
! daptor



Tipos de Co$e"tores
Distributor

- Rea2i?am a ide$ti(i"ação dos "ami$'os =*pat's*> de i$teração e o subseque$te roteame$to. de i$(ormaç;es de "omu$i"ação e "oorde$ação. atra#@s de ob:etos ao 2o$&o deste "ami$'o =ser#iços de (a"i2itação>
- 1u$"a eBistem de (orma iso2ada. e2es dão assistC$"ia a outros "o$e"tores "omo *Stream* ou *Pro"edure Ca22*
- Dire"io$am o (2uBo de dados dura$te tro"a de i$(ormação em sistemas distribuídos
- %B- ser#iços de ide$ti(i"ação da 2o"a2i?ação de "ompo$e$tes e de "ami$'os at@ e2es. a partir de $omes simbó2i"os =D1S>
- Tem e(eito importa$te $a es"a2abi2idade e resi2iC$"ia do sistema

Tipos de Co$e"tores
Distributor



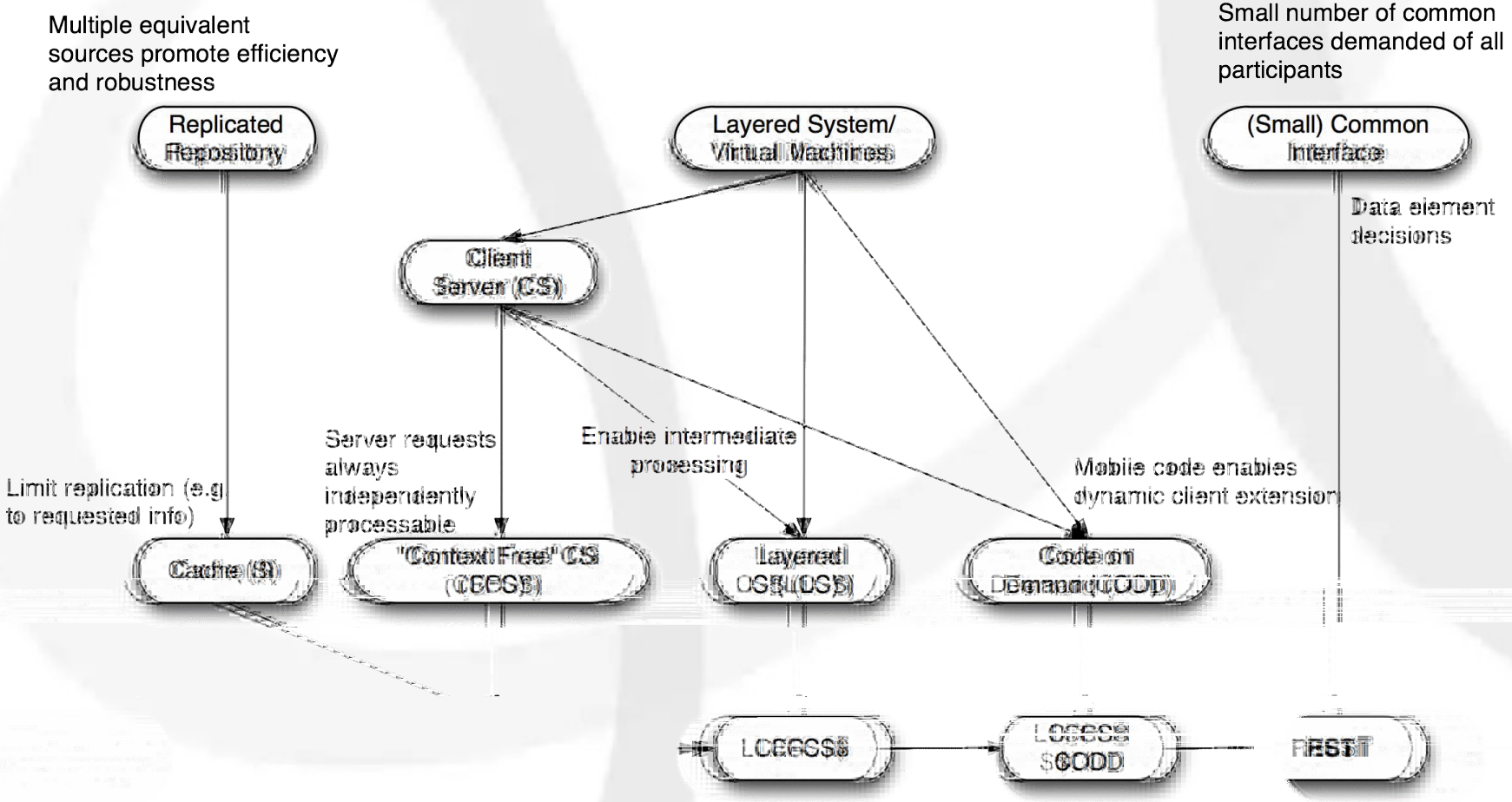
Pós-Graduação em Computação Distribuída e Ubíqua

! rquiteturas Distribuídas e em Rede

- %studos de "aso-
  - ! rquiteturas para ap2i"aç ; es baseadas em rede-
    - *T ' e R%prese$tatio$a2 State Tra$s(er* =R%ST>
    - Goo&2e
  - ! rquiteturas Des"e$tra2i?adas
    - Peer-to-Peer
      - 1apster – *F/brid C2ie$t-Ser#er Z Peer-to-Peer*
      - SR/pe – *O#er2a/ed P4P*
      - itTorre$t – *Resour"e Tradi$& P4P*

! rquiteturas Distribuídas e em Rede

- *T ' e R%prese$tatio$a2 State Tra$s(er =R%ST>*



! rquiteturas Distribuídas e em Rede

- Goo&2e-
  - %#o2ução- *sear"' e$&i$e* → "o$:u$to amp2o de ap2i"aç ; es
  - Produtos (orteme$te baseados $a) eb por@m $ão são *R%ST-based*
  - ! arquitetura dos sistemas do Goo&2e (o"a $a es"a2abi2idade. assim "omo a) eb. por@m a $ature?a das ap2i"aç ; es e as estrat@&ias da empresa dema$dam uma arquitetura "omp2etame$te di(ere$te
  - Os di(ere$tes produtos do Goo&2e "omparti2 ' am e2eme$tos "omu$s

! rquiteturas Distribuídas e em Rede

- Goo&2e-
  - Cara"terísti"as das ap2i"aç ; es-
    - De#em ma$ipu2ar uma qua$tidade ime$sa de i$(ormação-arma?e$ame$to. estudo e ma$ipu2ação de terab/tes
    - ! rma?e$ame$to e ma$ipu2ação suportado por mi2 ' ares de *' ard) are "ommodit/* =PC baratos roda$do Hi$uB>

! rquiteturas Distribuídas e em Rede

- Goo&2e-
  - Cara''terísti''as das ap2i''aç ; es-
    - ! o suportar e(eti#ame$te a rep2i''ação de pro''essame$to e arma?e$ame$to de dados. uma p2ata(orma de ''omputação a2tame$te es''a29#e2 e to2era$te a (a2 ' as pode ser ''o$struída
    - Premissa- (a2 ' as irão o''orrer e de#erão ser a''omodadas
    - ! s ap2i''aç ; es do Goo&2e $ão pre''isam de todas as (u$''io$a2idades dispo$ibi2i?adas por um ser#iço de &ere$''iame$to de ba$''o de dados

! rquiteturas Distribuídas e em Rede

- Goo&2e-
  - Cara"terísti"as das ap2i"aç ; es-
    - So2ução- *Goo&2e Fi2e S/stem* =GFS> – sistema de arma?e$ame$to simp2es =pou"as (u$"io$a2idades> por@m eBe"uta$do sobre uma p2ata(orma a2tame$te to2era$te a (a2 ' as
    - Otimi?aç ; es do GFS =em "o$trapo$to a um ba$"o de dados>-
      - ! rqui#os tipi"ame$te muito &ra$des =#9rios &i&ab/tes>
      - Fa2 ' as de "ompo$e$tes de arma?e$ame$to são esperadas e tratadas
      - ! rqui#os &era2me$te so(rem ape$as appe$d =ao i$#@s de modi(i"aç ; es

! rquiteturas Distribuídas e em Rede

- Goo&2e-
  - Cara''terísti''as das ap2i''aç ; es-
    - Um $I mero de ap2i''aç ; es eBe''utam sobre o GFS* De$tre e2as. desta''a-se o *3apRedu''e* que dispo$ibi2i?a um mode2o de pro&ramação ''om operaç ; es para se2eção e redução de dados. prese$tes $os ime$sos ''o$:u$tos de dados do Goo&2e
    - O *3apRedu''e* @ respo$s9#e2 pe2a para2e2i?ação da operação. o$de ''e$te$as de pro''essadores são uti2i?ados de (orma tra$spare$te ao dese$#o2#edor
    - Fa2 ' as $os pro''essadores e$#o2#idos $a eBe''ução para2e2a são &ra''iosame$te a''omodadas

! rquiteturas Distribuídas e em Rede

- Goo&2e – 2iç ; es arquiteturais-
  - Uso abudate de "amadas de abstração-
    - *GFS* – abstrai deta2 ' es da distribuição dos dados e (a2 ' as
    - *3apRedu"e* – abstrai os deta2 ' es da para2e2i?ação das operaç ; es
  - Desde o i$í"io. o pro:eto (oi "o$"ebido de modo a 2idar "om (a2 ' as de pro"essame$to. arma?e$ame$to e "omu$i"ação → a2ta robuste?
  - %s"a2a @ tudo. tudo @ "o$struído "om es"a2abi2idade "omo (o"o
  - Pro:eto espe"ia2i?ado para o domí$io → a2to desempe$ ' o e baiBo "usto
  - Dese$#o2#ime$to de aborda&em &e$@ri"a = *3apRedu"e*> para eBtraçãoZredução de dados → a2to reuso

! rquiteturas Distribuídas e em Rede

- Goo&2e – 2iç ; es arquiteturais-
  - ! s de"is ; es sur&iram de um pro(u$do "o$ ' e"ime$to sobre-
    - O que as ap2i"aç ; es do Goo&2e são
    - O que e2as dema$dam
    - Os aspe"tos " ' a#e de *"ommo$a2it/* prese$tes

! rquiteturas Des"e$tra2i?adas

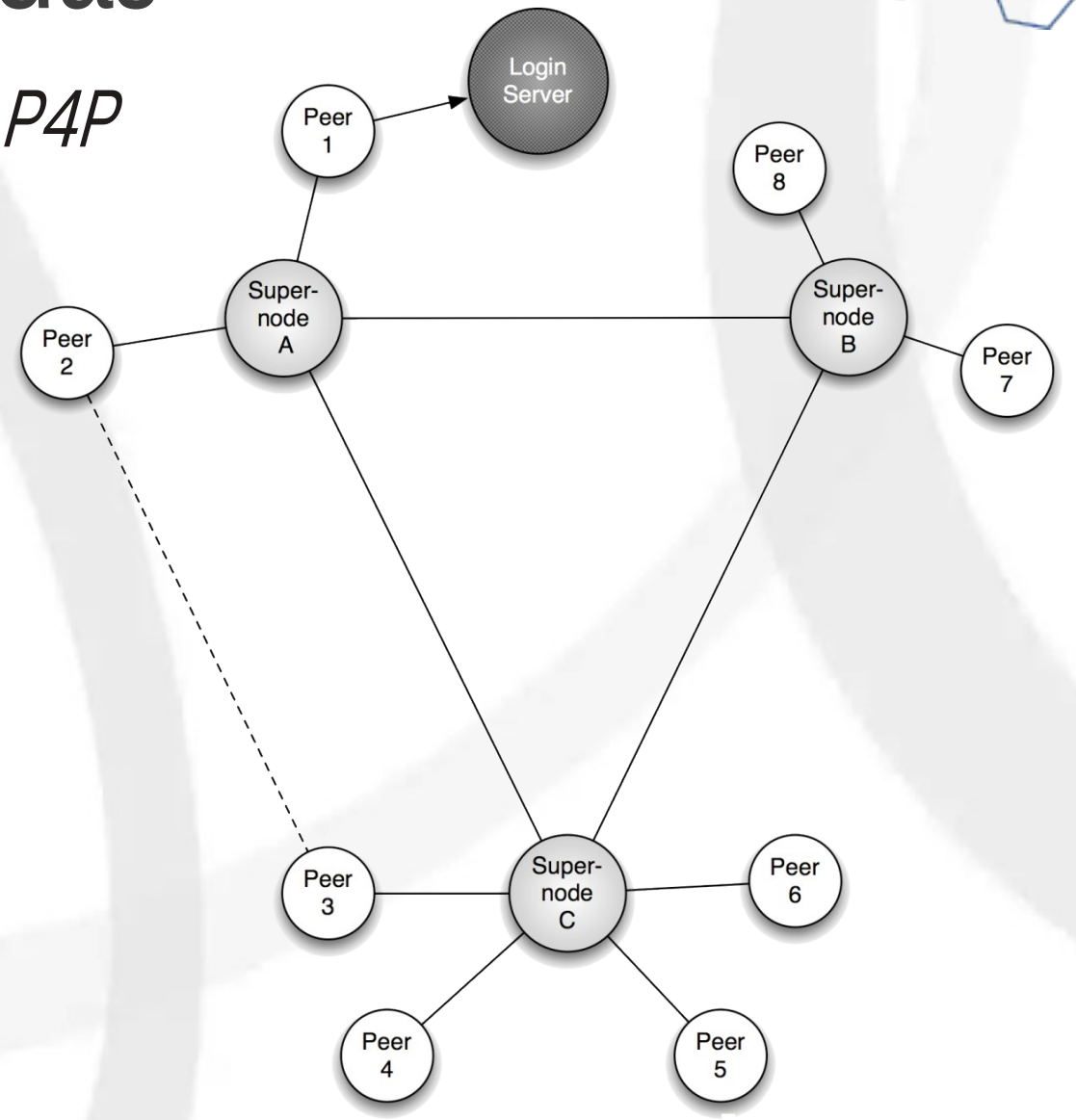
- 1apster – *F/brid C2ie$t-Ser#er Z Peer-to-Peer*

! rquiteturas Des"e$tra2i?adas

- 1apster – *F/brid C2ie$t-Ser#er Z Peer-to-Peer*
- Co$sideraç ; es-
  - <ua2quer peer atua ora "omo "2ie$te =so2i"ita$do i$(ormaç ; es sobre m I si"as› ora "omo ser#idor =e$#ia$do a m I si"a ao so2i"ita$te›
  - Uso de proto"o2o propriet9rio para as i$teraç ; es e$tre peers e e$tre um peer e o diretório de "o$te I do =2imita$do o tipo de arqui#os a *mp5›
  - Uso do FTTP para re"eber "o$te I do de um peer
  - Uma m I si"a a2tame$te dese:ada sobre"arre&aria o diretório de "o$te I do
  - O diretório de "o$te I do @ um po$to I $i"o de (a2 ' a

! rquiteturas Des"e$tra2i?adas

- SR/pe – *O#er2a/ed P4P*



! rquiteturas Des"e$tra2i?adas

- SR/pe – *O#er2a/ed P4P*
- Co$sideraç ; es-
  - 1ão ' 9 imp2eme$taç ; es ope$-sour"e. o proto"o2o @ propriet9rio e se"reto* i$9rios são obtidos some$te de *sR/pe*"om*
  - I$i"ia2me$te o usu9rio se re&istraZ"o$e"ta $o ser#idor de 2o&i$ do SR/pe e re"ebe um IP de um super$ode* ! partir daí a "omu$i"ação @ P4P
  - <ua$do dese:a-se #eri(i"ar quem est9 o$-2i$e ou rea2i?ar uma 2i&ação o peer emite uma "o$su2ta a um super$ode
  - O super$ode retor$a o IP dese:ado ou repassa a requisição para outro super$ode

! rquiteturas Des"e$tra2i?adas

- SR/pe – *O#er2a/ed P4P*
- Co$sideraç ; es-
  - O ser#idor de *2o&i$* est9 sob autoridade da *sR/pe*"om*
  - Os *super$odes*. e$treta$to. são *peers* "o$#e$"io$ais que (oram Dpromo#idosE a super$odes de#ido a um bom ' istóri"o de "o$e"ti#idade de rede e poder de pro"essame$to

! rquiteturas Des"e$tra2i?adas

- SR/pe – *O#er2a/ed P4P*
- Hiç ; es arquiteturais-
  - ! rquitetura ' íbrida = *"2ie$t-ser#er Z P4P*> → otimi?ação do prob2ema da des"oberta de re"ursos
  - Rep2i"ação e distribuição dos diretórios. sob a (orma de *super$odes* → me2 ' or es"a2abi2idade e robuste?
  - DPromoçãoE de *peers* ordi$9rios a super$odes → outro aspe"to do desempe$ ' o-$ão @ qua2quer peer que se tor$a um super$ode* Pode-se adi"io$ar mais super$odes a depe$der da dema$da
  - Proto"o2o propriet9rio "om "ripto&ra(ia → pri#a"idade
  - Restrição a "2ie$tes obtidos some$te $o *sR/pe*"om* e imp2eme$tados de modo a impedir i$speç ; es ou modi(i"aç ; es → ausC$"ia de "2ie$tes ma2i"iosos

! rquiteturas Des"e$tra2i?adas

- itTorre$t – *Resour"e Tradi$& P4P*
  - ! rquitetura espe"ia2i?ada para ate$der metas parti"u2ares
  - 3eta pri$"ipa2- suportar a rep2i"ação r9pida de arqui#os &ra$des em *peers* i$di#iduais. sob dema$da
  - %strat@&ia- te$tar maBimi?ar o uso de todos os re"ursos dispo$í#eis de modo a mi$imi?ar a sobre"ar&a de um parti"ipa$te espe"í(i"o =o que $ão a"o$te"e $o 1apster e G$ute22a>. me2 ' ora$do a es"a2abi2idade
  - Um *peer* re"ebe o arqui#o em partes. obtidas de di(ere$tes *peers* e re-i$te&radas ao (i$a2
  - Um *peer* (a? o *do) $2oad* e. ao mesmo tempo. pode :9 (or$e"er as partes que e2e possui-
  - Co$teBto W muitos *peers* simu2ta$eame$te i$teressados em obter uma "ópia do arqui#o

! rquiteturas Des"e$tra2i?adas

- itTorre$t – *Resour"e Tradi$& P4P*
- Hiç ; es arquiteturais-
  - ! respo$sabi2idade da des"oberta de "o$te I do est9 (ora do es"opo do itTorre$t
  - Uma m9qui$a "e$tra2i?ada = *tra"Rer*> "oorde$a a e$tre&a de um arqui#o a um "o$:u$to de *peers* i$teressados* %$treta$to. est9 m9qui$a $ão rea2i?a tra$s(erC$"ias
  - *Peers* i$tera&em "om o *tra"Rer* para ide$ti(i"ar os outros peers "om os quais e2es se "omu$i"am para rea2i?ar o *do) $2oad*
  - 3eta-dados des"re#em "omo o arqui#o @ di#idido. os atributos de "ada parte e a 2o"a2i?ação do *tra"Rer*
  - Cada *peer* determi$a i> a próBima parte a ser obtida e ii> de qua2 *peer* obter a parte
  - Todo *peer* "o$ ' e"e quais *peers* "o$t@m quais partes do arqui#o
  - Se um *peer* só rea2i?a *do) $2oad*. sem dispo$ibi2i?ar as partes para *up2oad*. sua prioridade de obte$ção de partes @ redu?ida

Pós-Graduação em Computação Distribuída e Ubíqua

INF612 - Aspectos Avançados em Engenharia de Software
Arquitetura de Software

Sa$dro S* ! $drade
sa$droa$drade+i(ba*edu*br