

- Final: take-home (like prelim)
 - Available: 13 May
 - Due: 20 May *firm*
- Project presentations:
 - 10:00, 1:00, 3:00 weekdays
 - between 3:00 6 May and 3:00 19 May
 - Signup sheet Kathy, 4115 Upson

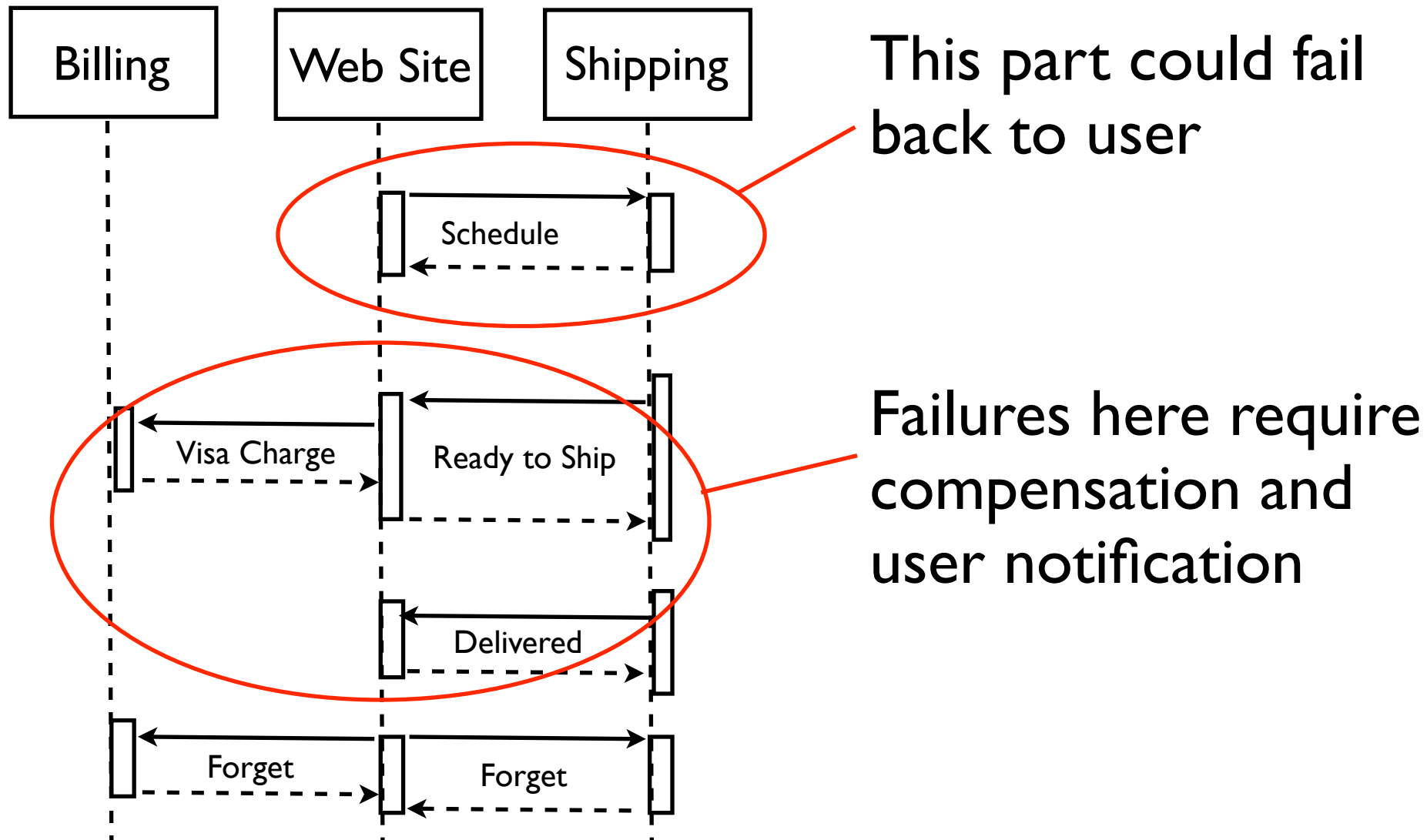
Failures

- How detect them?
- Recovery at the client?

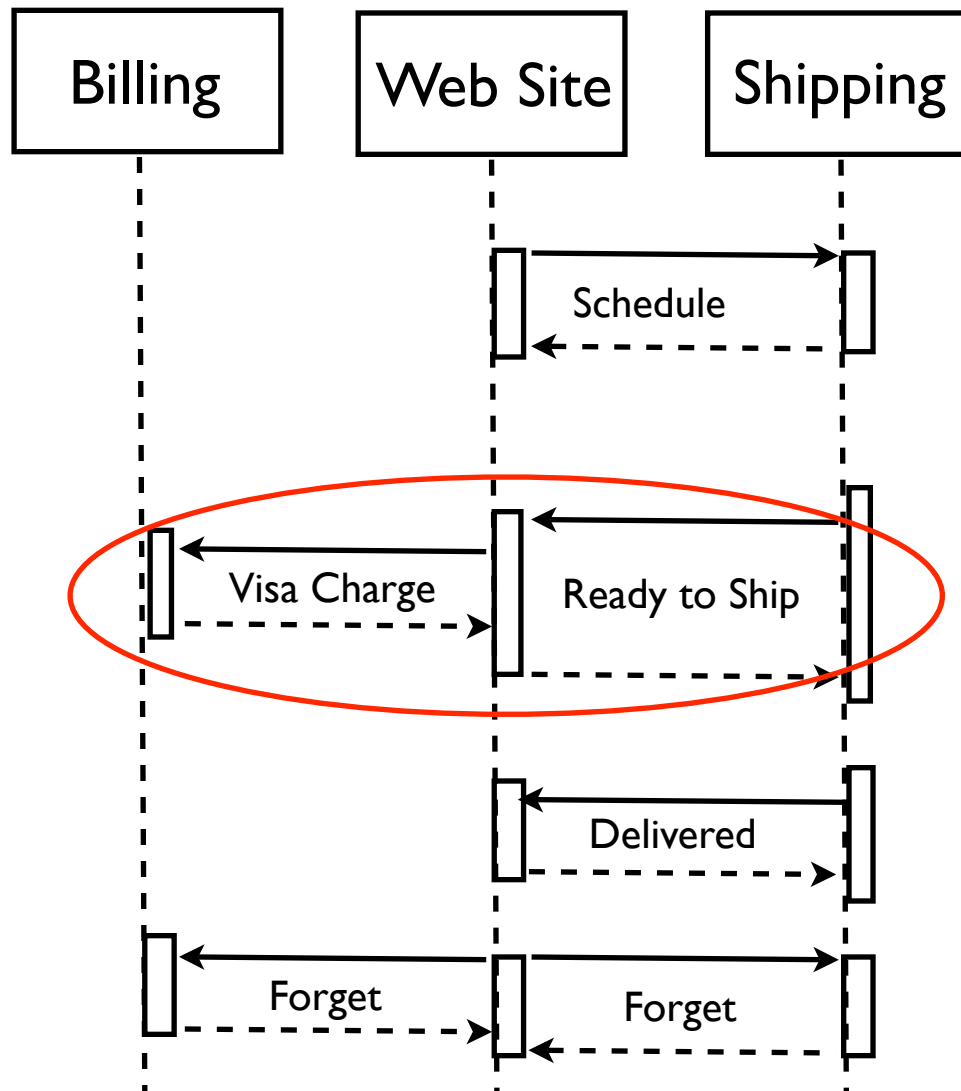
Client View of Server Failure

- A synchronous call will timeout and raise an exception immediately
- MOM-managed messages will put messages in a queue and retry delivery to receiving server(s)
 - Multiple delivery failures => error log?
 - Failure to process correctly => compensate (app's responsibility)

Shipping and Billing Again ...



Async Failure



Failure at Web or Billing sites raises exception at Shipping site

Could *not* do compensating rollback at this point!

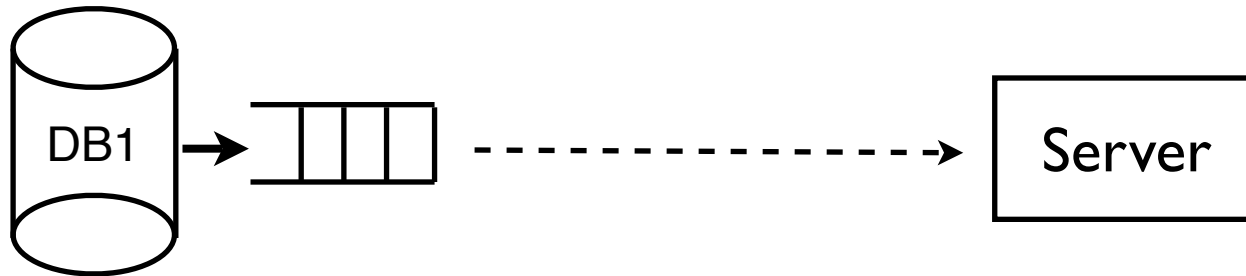
Site needs long-term retry mechanism

- The fact that a business op is retrying must be persistent (ACID txn required)
- Retries should respect original order if app is at all sensitive to this!
- Failure at this level => unrecoverable error
- Error log for manual recovery

User Notification

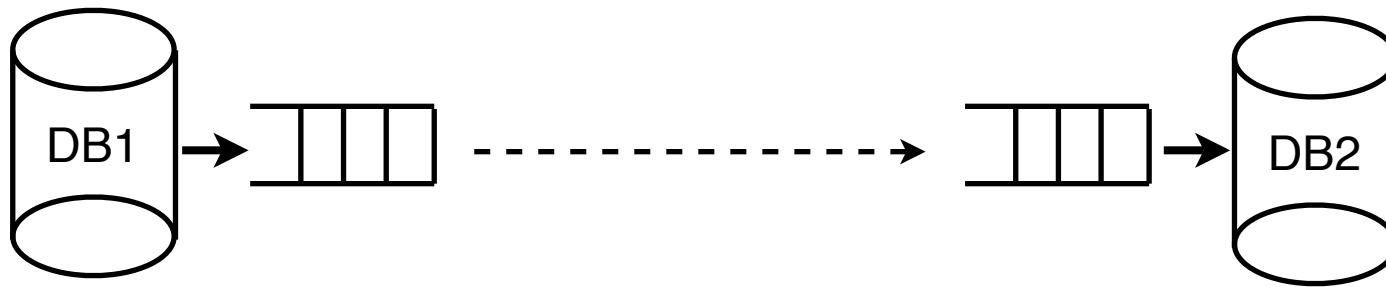
- Even “real” e-Business sites notify customers of exceptions by email
- But eMail is *not* transactional!
- Oh, well ...

Manual Queue



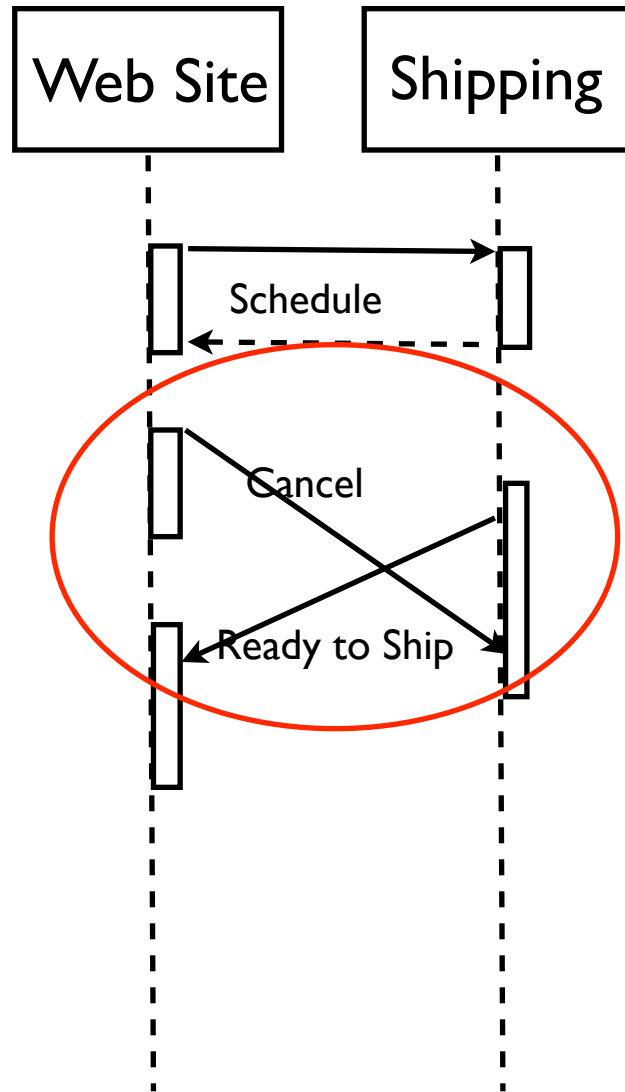
- Queue is a relational table in DB
- Queue items timestamped
- Daemon process to drain queue
- Daemon uses SQL for dequeue
 - rely on server for txn behavior and dup suppression
- Cannot preserve queue ordering if concurrent calls allowed

Bulk Queue Transfer



- Queue is a relational table in *each* DB
- Queue propagation uses distributed txns if available to avoid duplication of queue items
 - not critical if dup elimination is done
- Apply process entirely on server side
- Preserve queue ordering in apply?

Async User Request



What if the messages cross?

Care required to avoid deadlock

Care required for consistent resolution of conflict at both sides