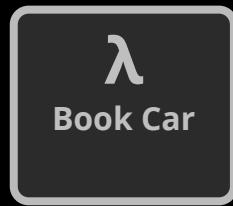


Coordination of Serverless Functions

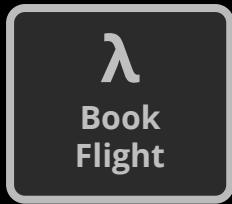
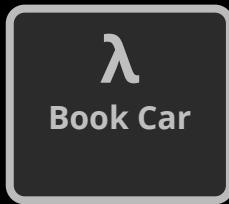
@berndruecker



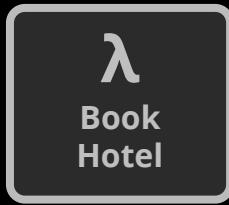
An example for today – the classical one



Function collaboration

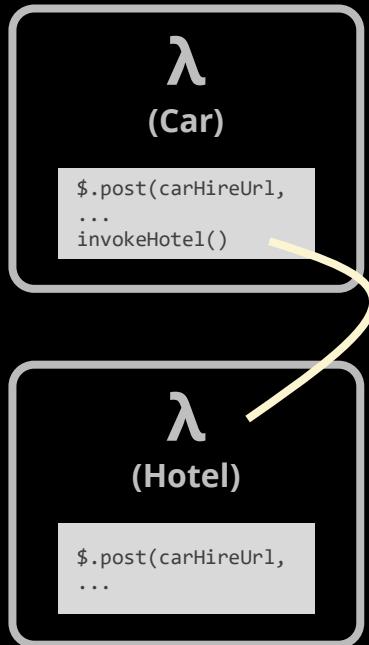


Book a
trip?

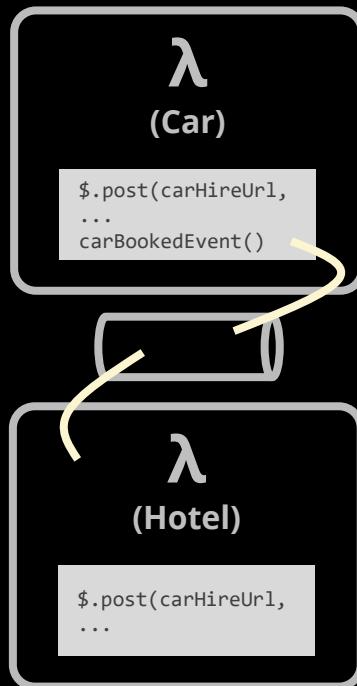


Choreography

Chaining



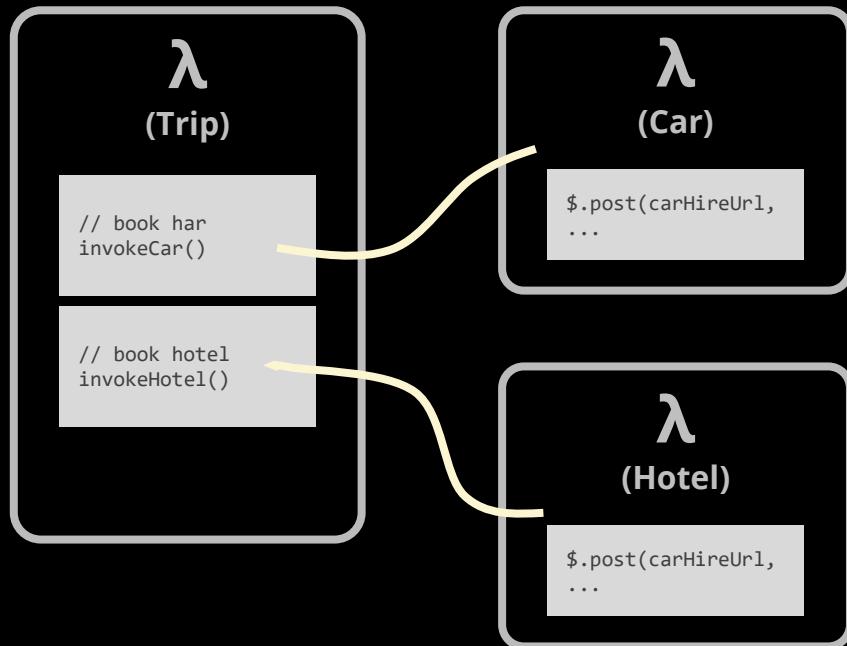
Event-Chain



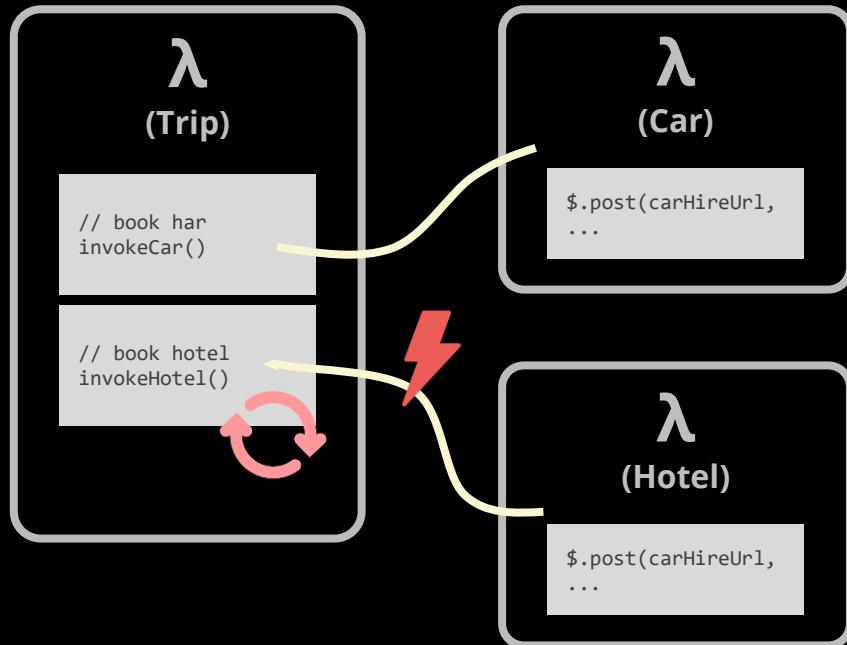
No extra framework needed

The flow is scattered across functions and thus invisible
Changing the sequence is hard

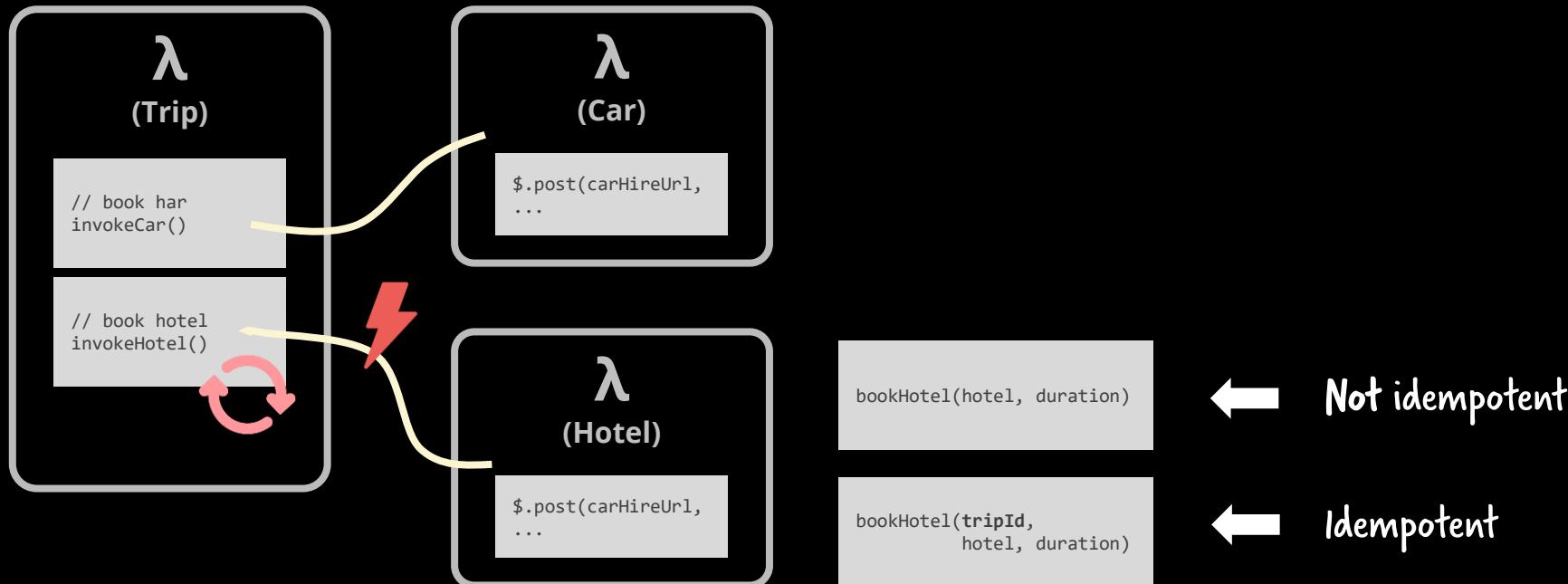
orchestration / coordinator function



orchestration / coordinator function

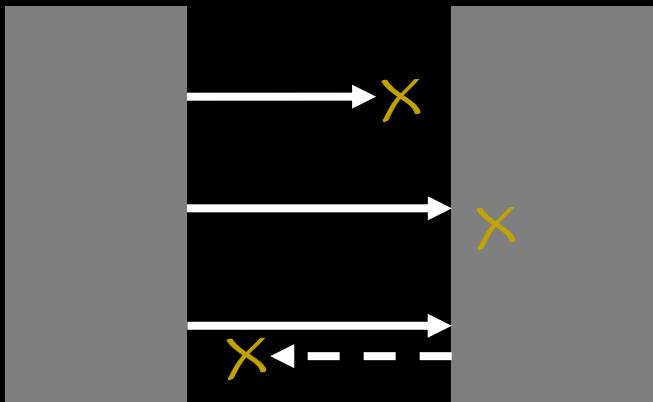


orchestration / coordinator function

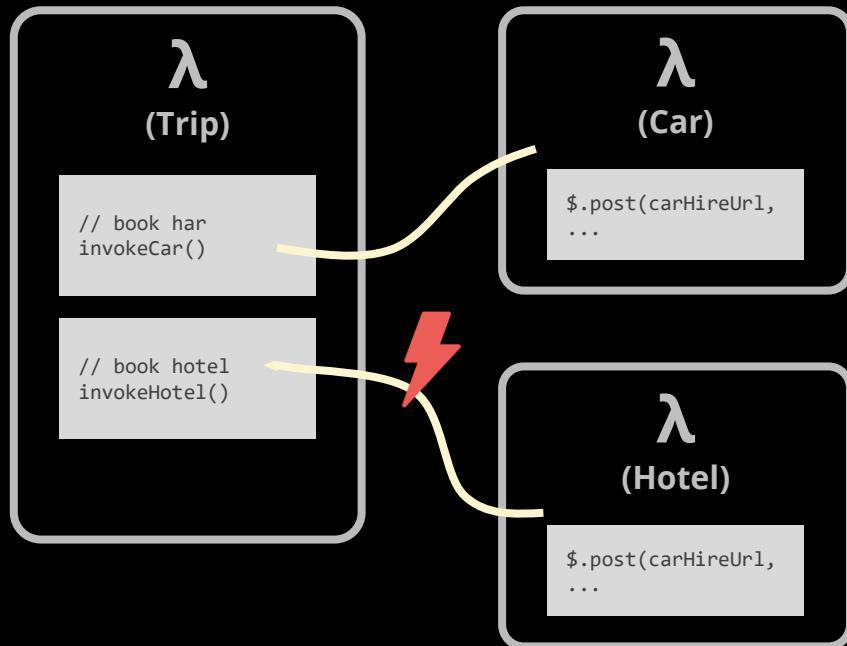


It is impossible to
differentiate certain
failure scenarios:

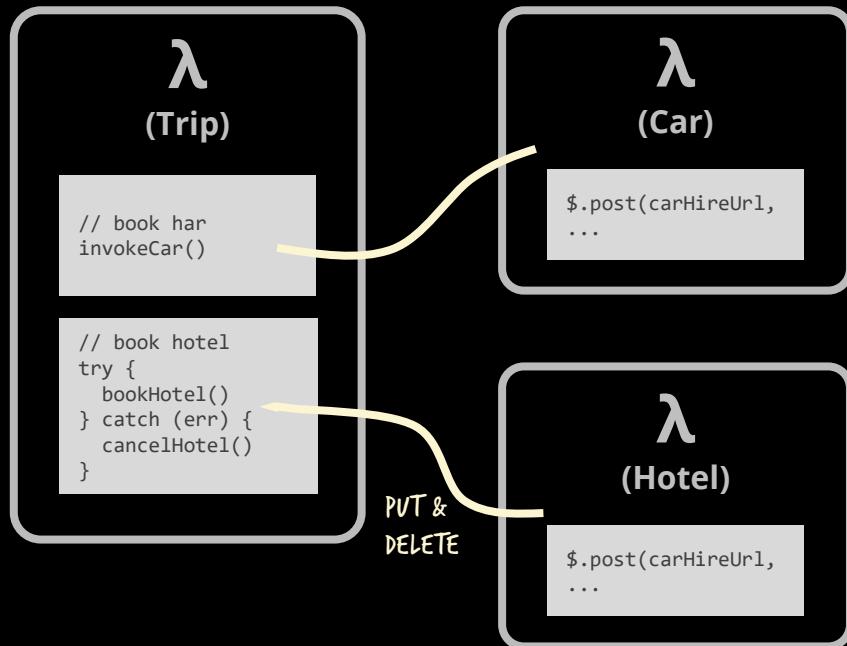
Independant of
communication style!



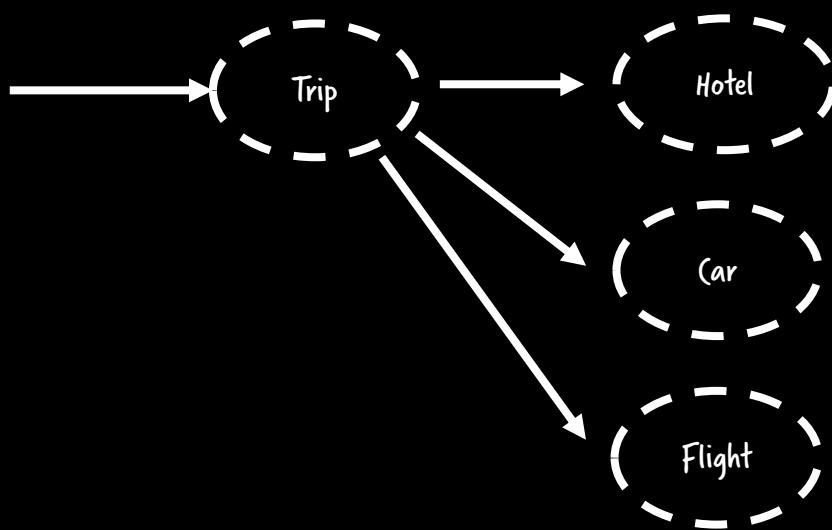
orchestration / coordinator function



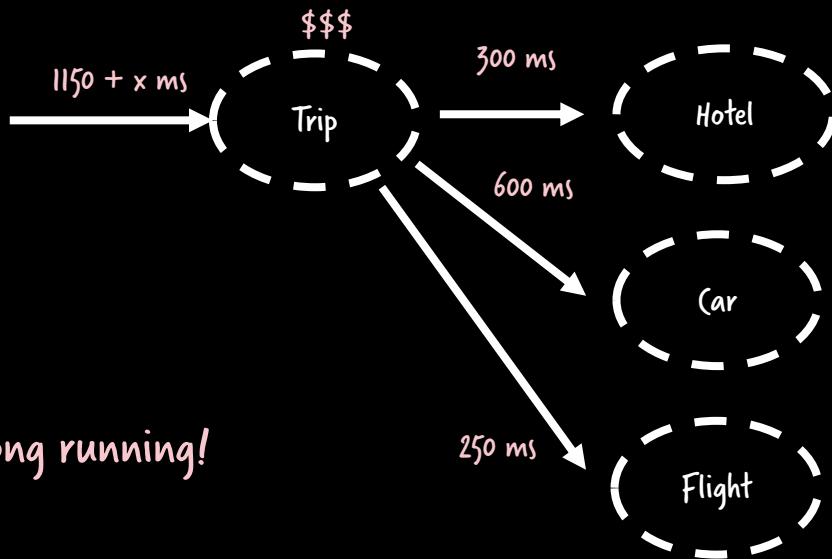
(cleanup!



Weaknesses

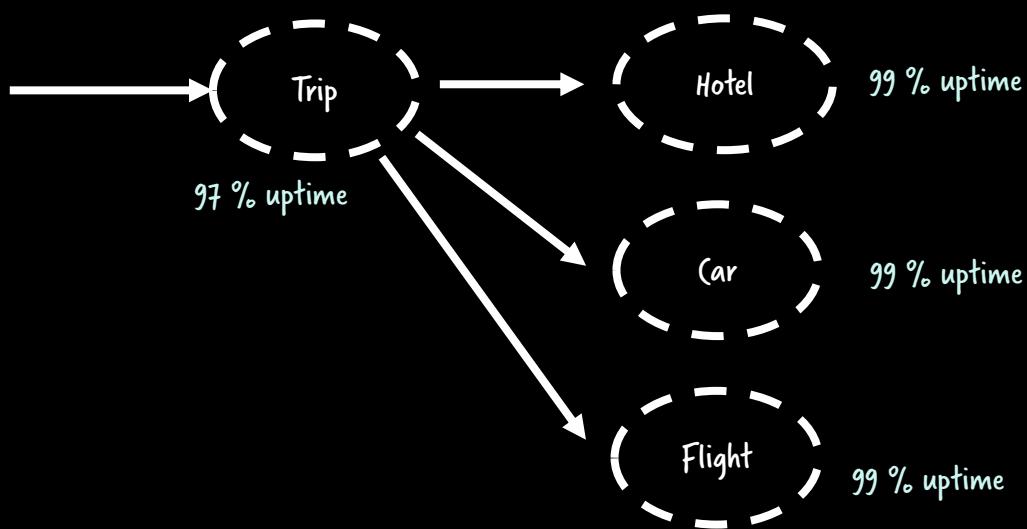


Weaknesses: Latency creep



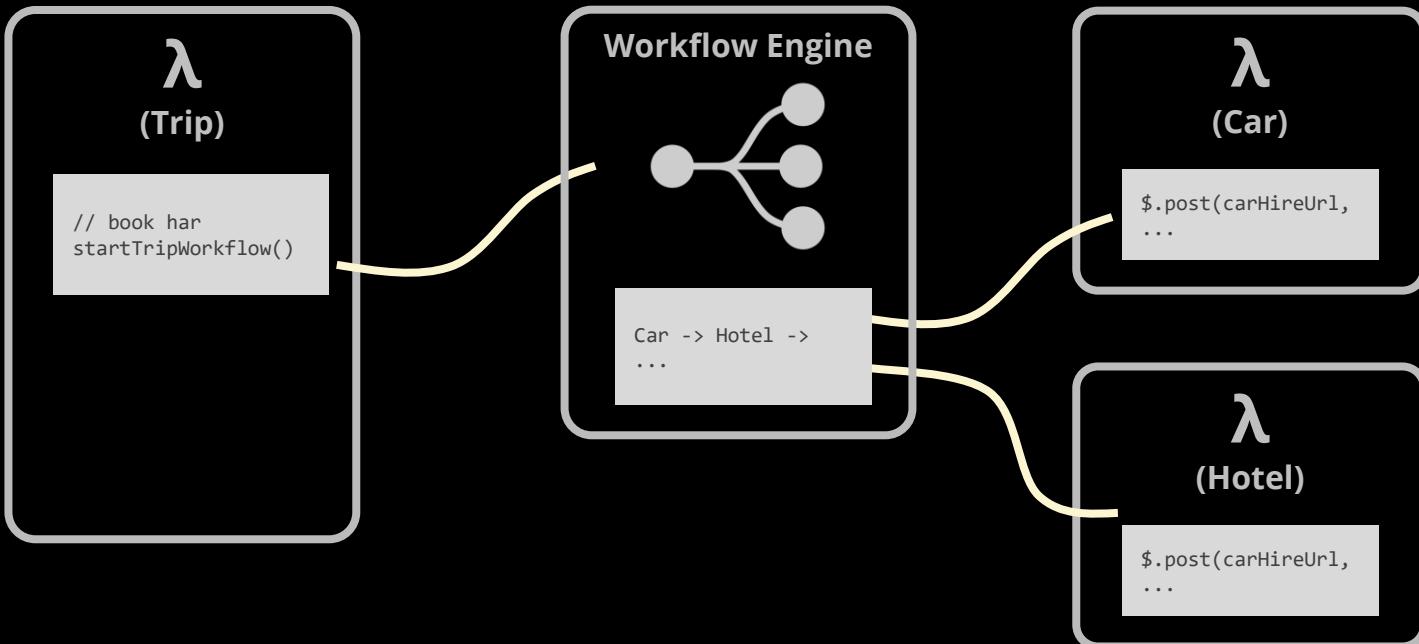
Coordinator get's long running!

Weaknesses: Availability erosion



Orchestration with workflow

Functions keep independant
No long running function
Extra component involved

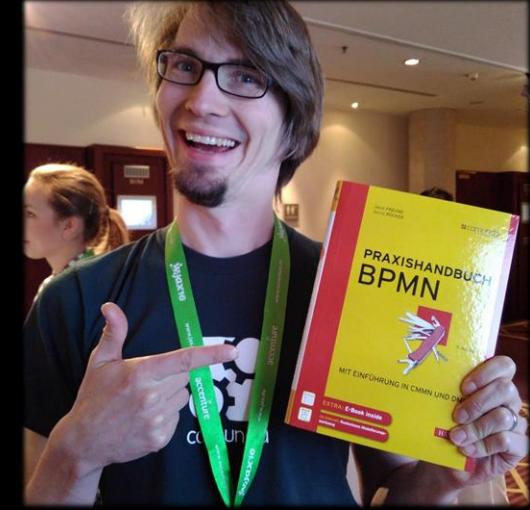


**Warning:
Contains Opinion**



Bernd Ruecker

Co-founder and
Chief Technologist of
Camunda

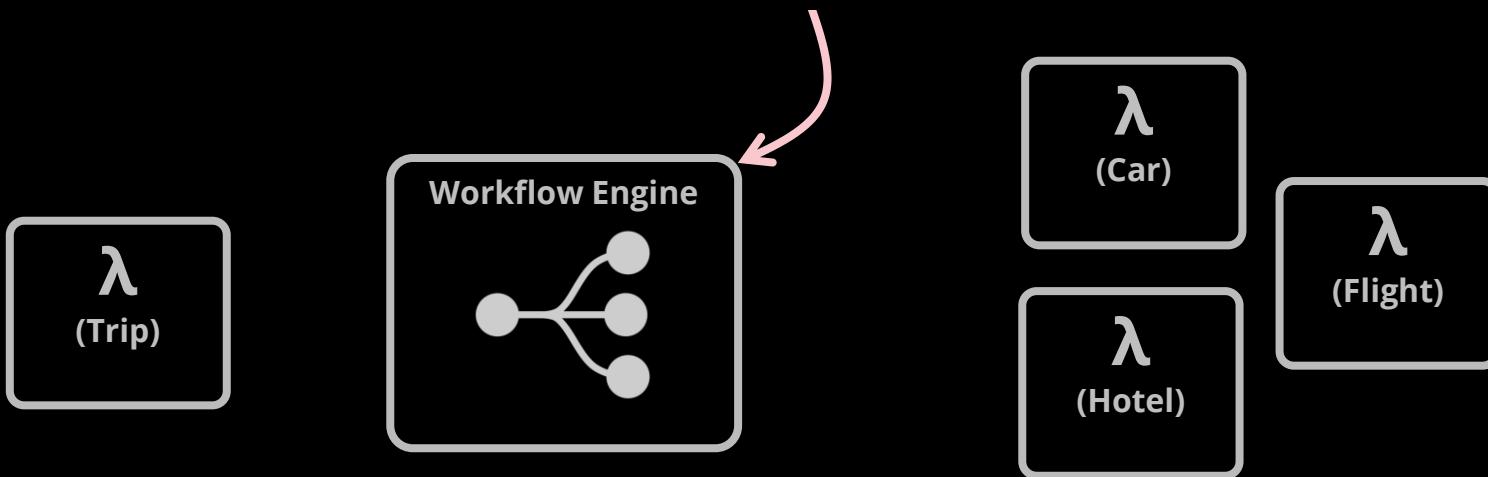
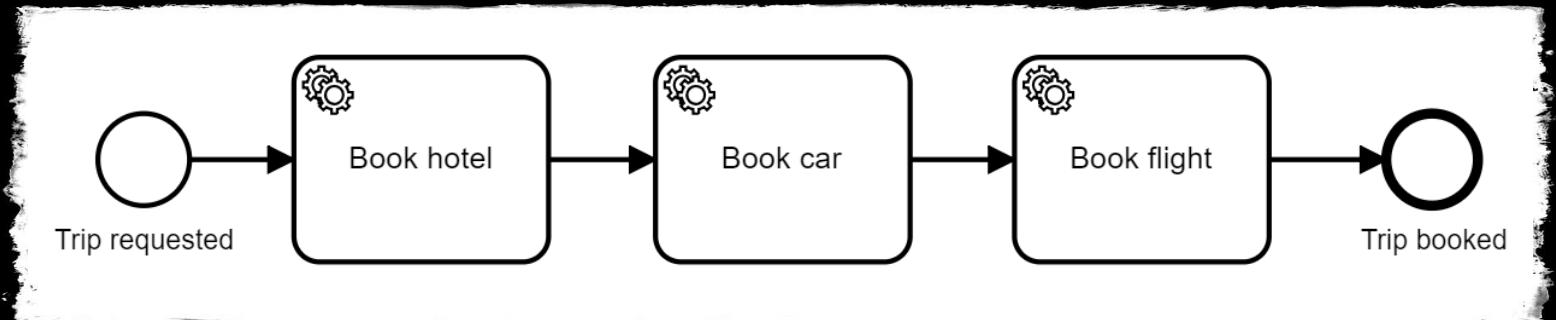


Berlin, Germany

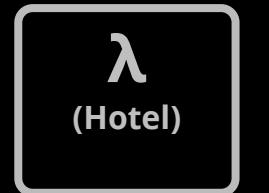
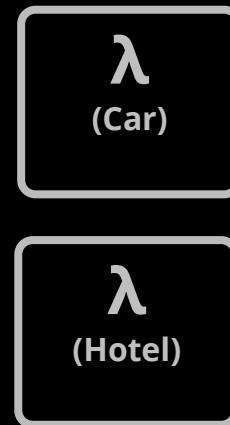
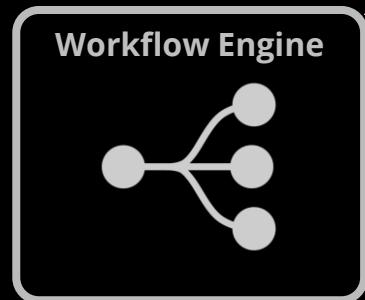
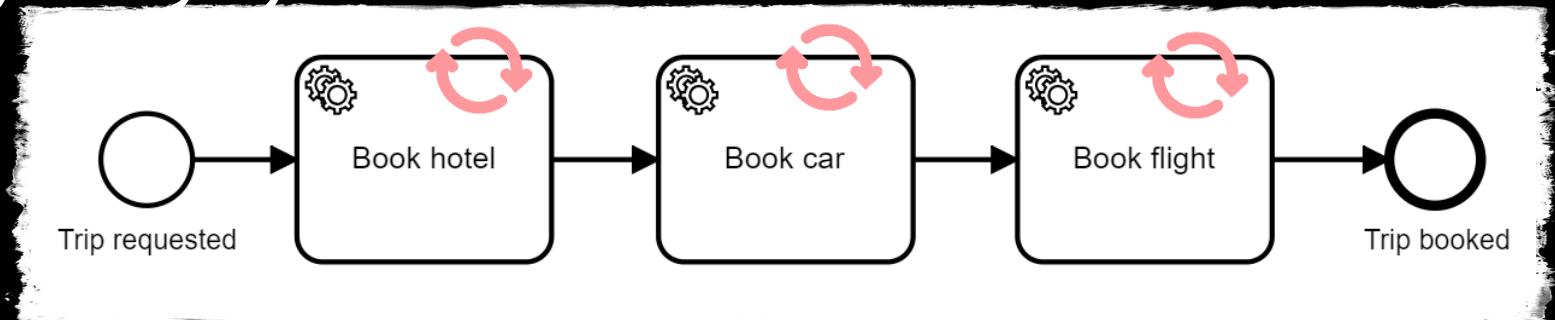
bernd.ruecker@camunda.com
[@berndruecker](https://twitter.com/berndruecker)



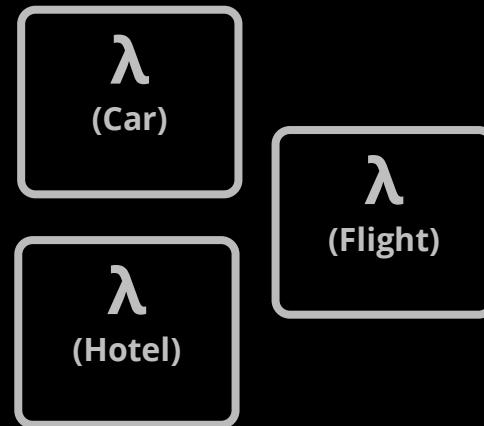
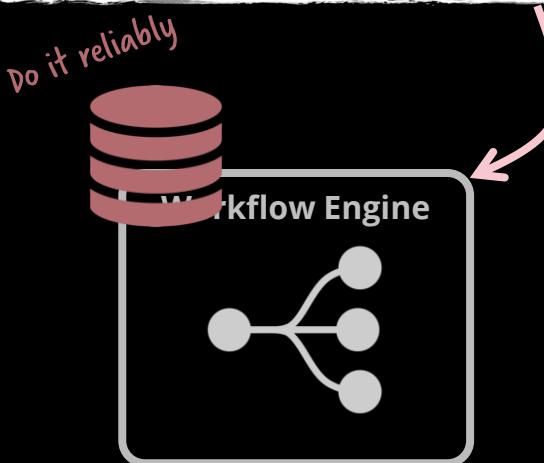
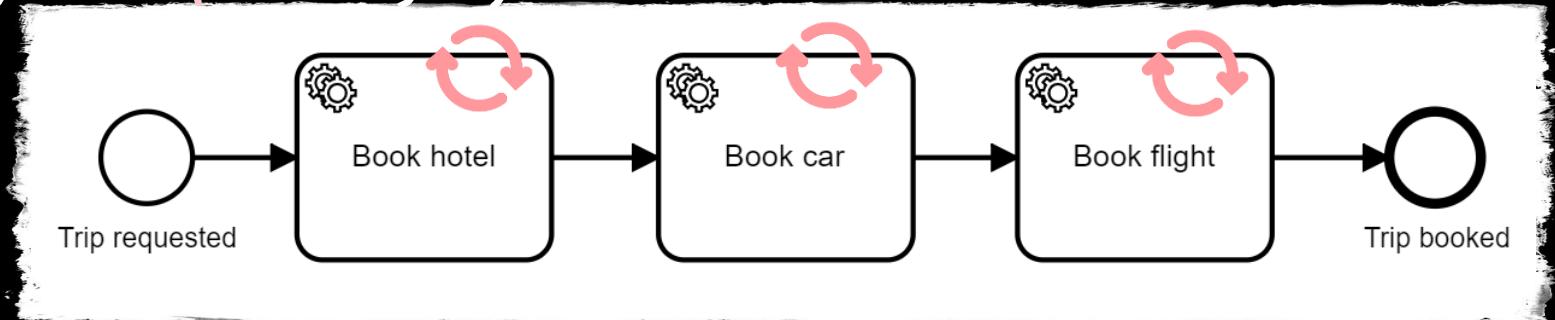
Example



Leverage retrying

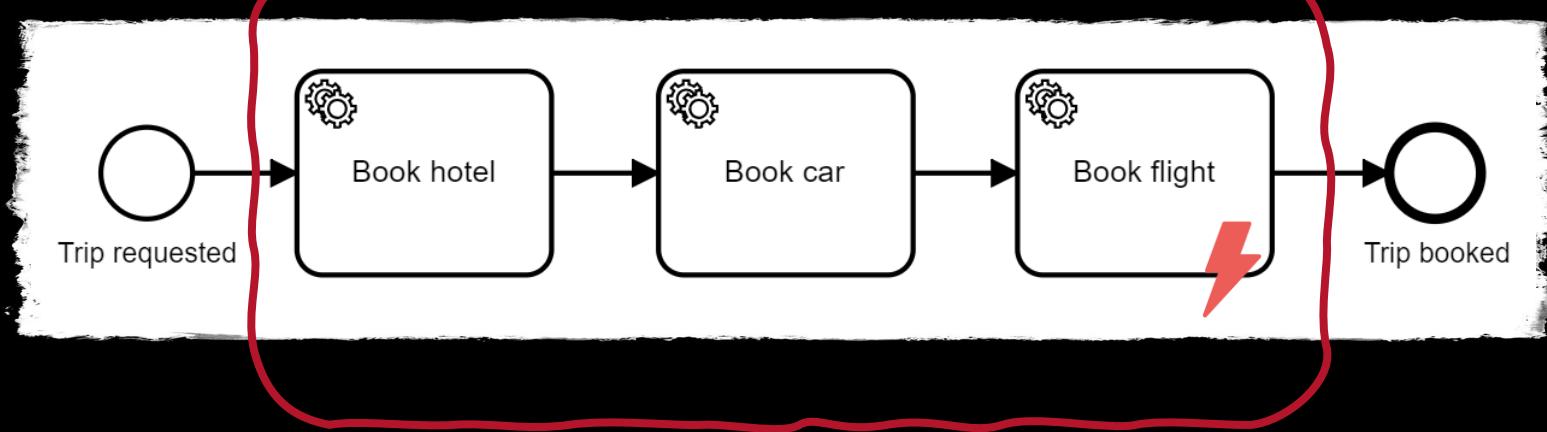


Leverage stateful retrying

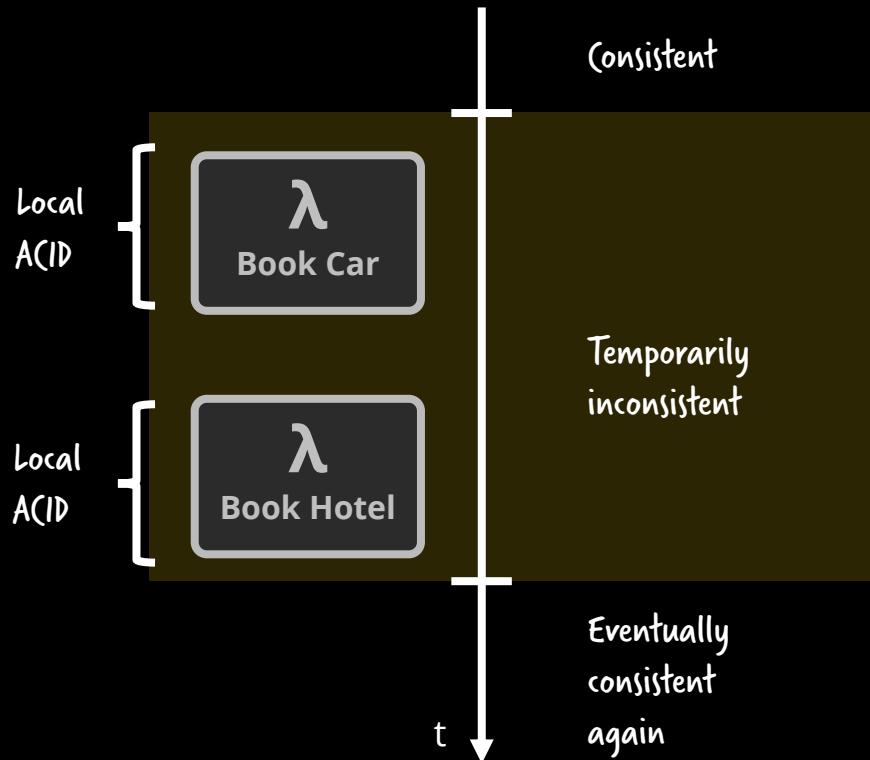


Let's talk about consistency

All or nothing

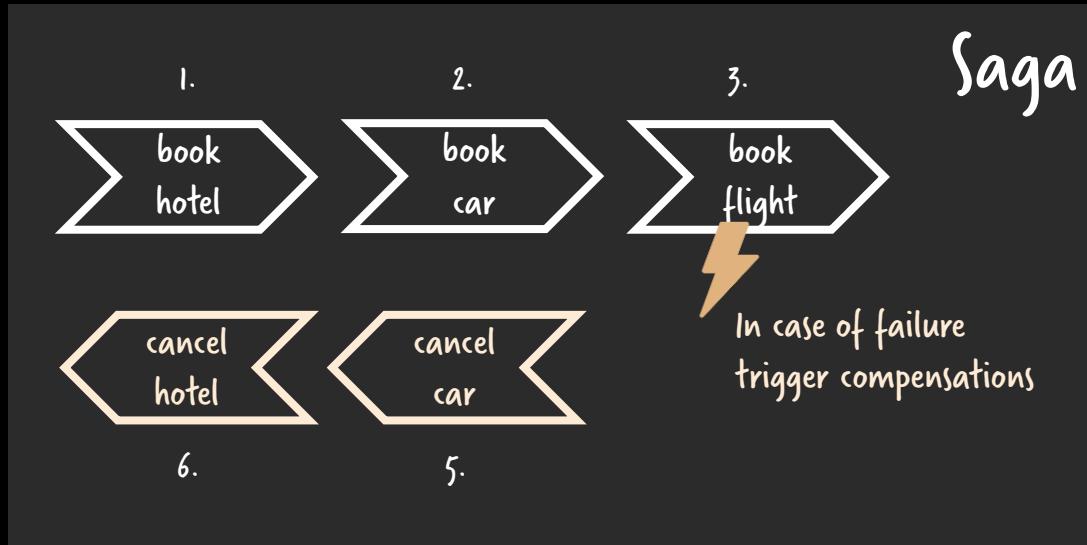


That means



Compensation – the classical example

→
book
trip



CAMUNDA

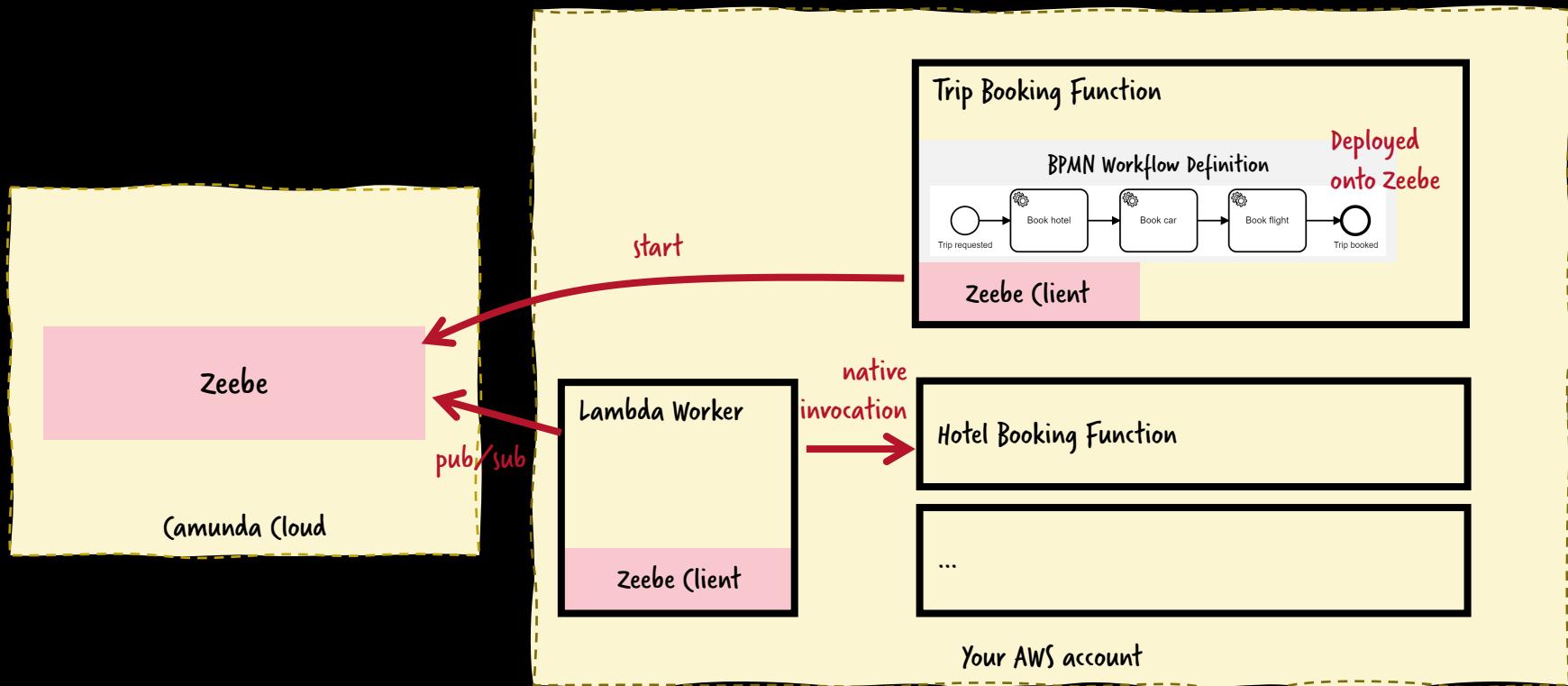
Camunda Cloud – Process Automation as a Service
Workflow engine based on BPMN
Cloud-native, Source-available, horizontally scalable

Live hacking



<https://github.com/berndruecker/trip-booking-saga-serverless>

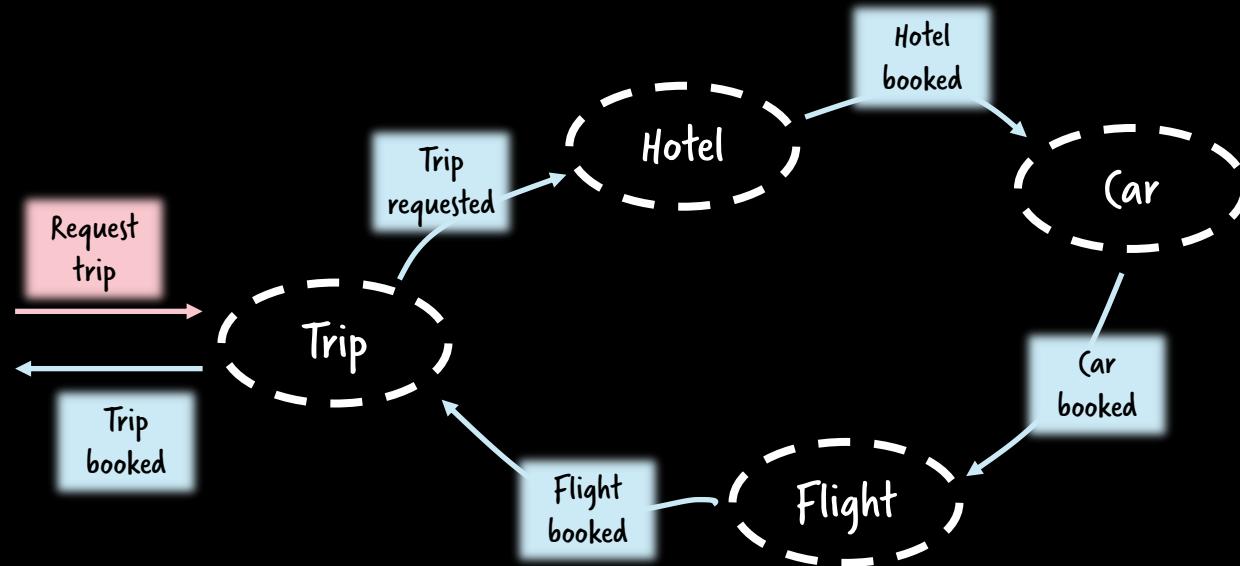
Demo Architecture



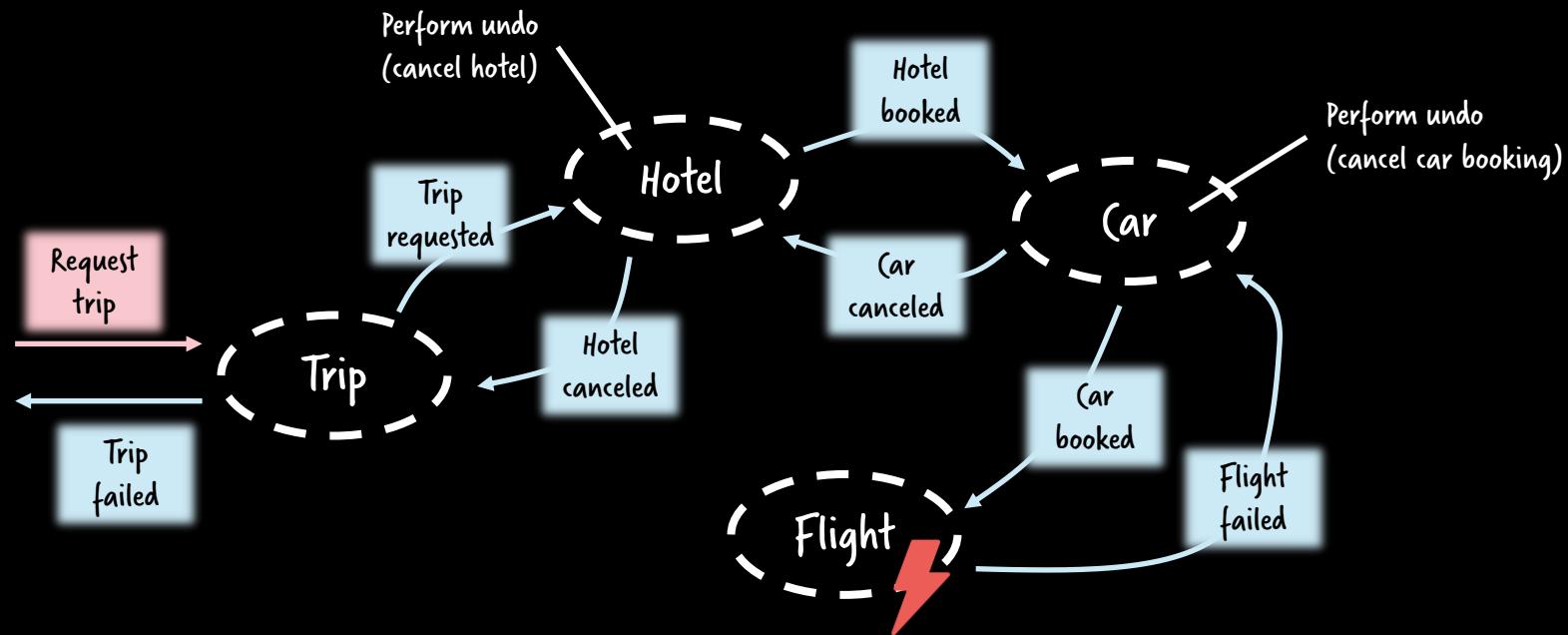
(choreography & orchestration



Event-driven choreography



Event-driven choreography





The danger is that it's very easy to make nicely decoupled systems with event notification, without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.

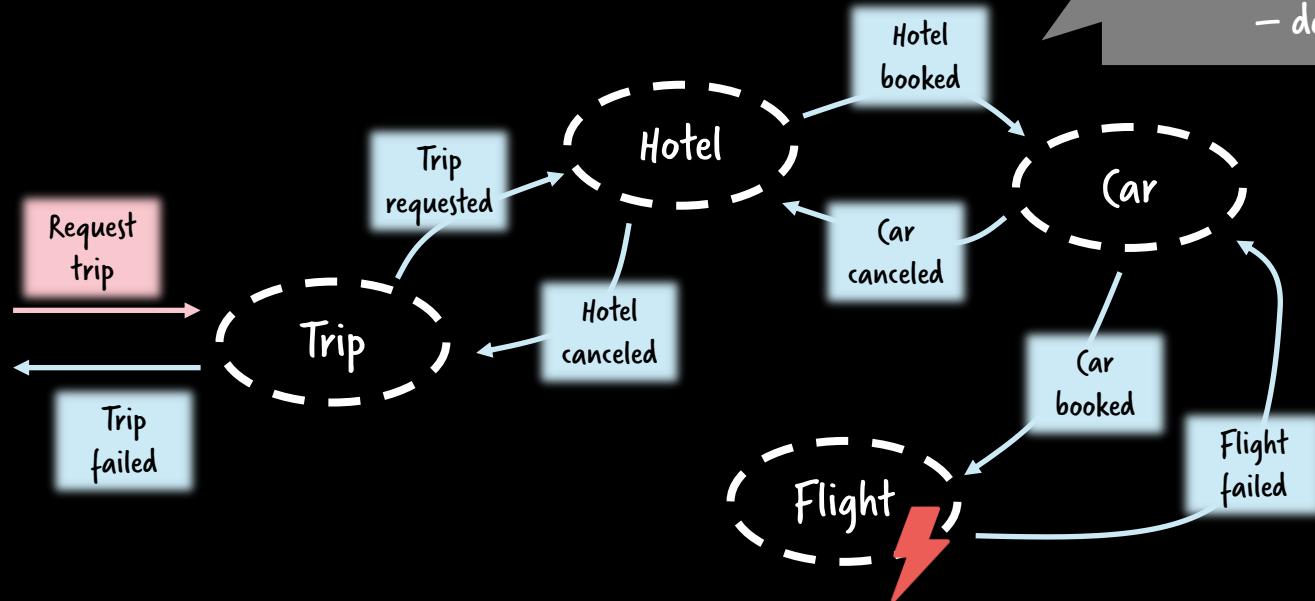


The danger is that it's very easy to make nicely decoupled systems with event notification, without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.



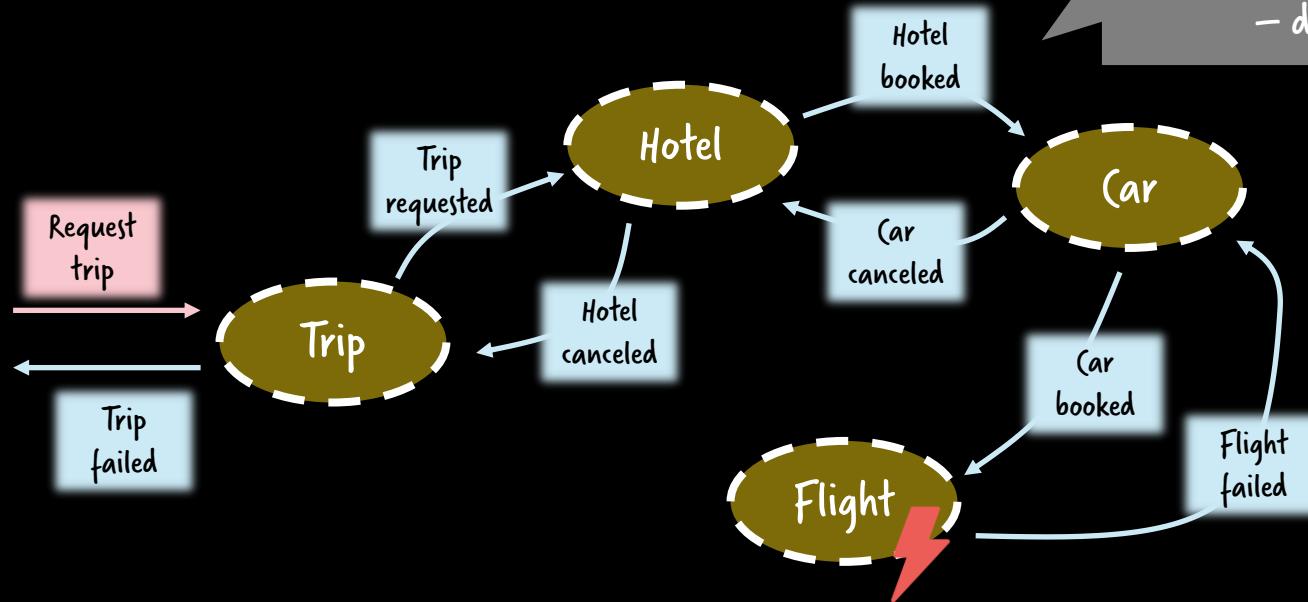
The danger is that it's very easy to make nicely decoupled systems with event notification, without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.

Implementing changes in the process



We have a new basic agreement
with the car rental agency and
can cancel for free within 1 hour
– do that first!

Implementing changes in the process



We have a new basic agreement
with the car rental agency and
can cancel for free within 1 hour
– do that first!

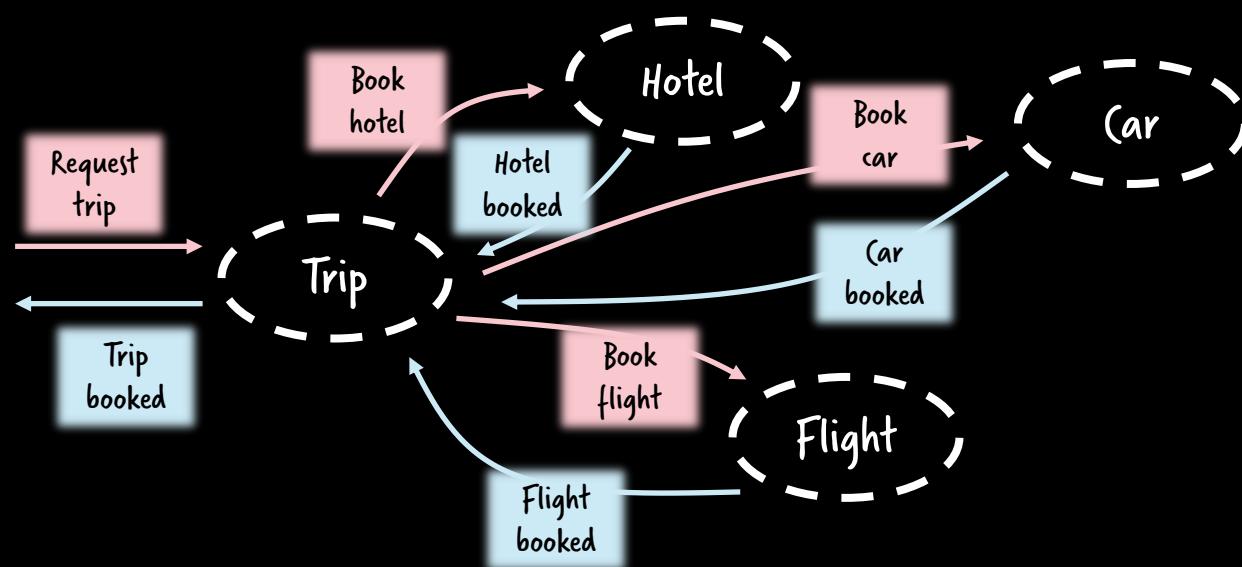
You have to adjust all services and redeploy at the same time!

What we wanted

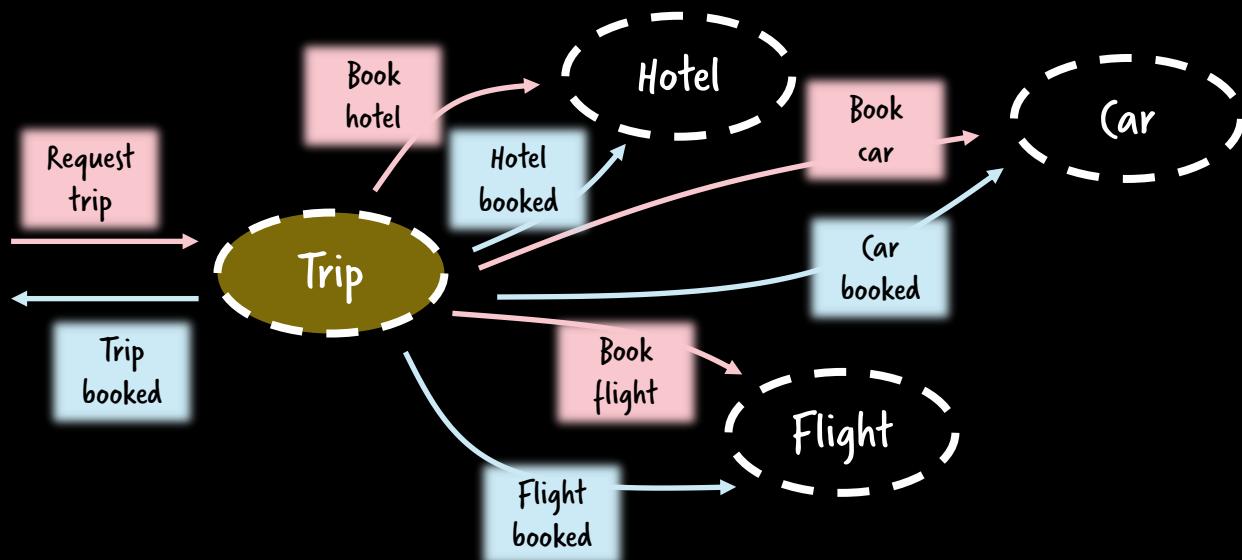


vs. what we got

orchestration



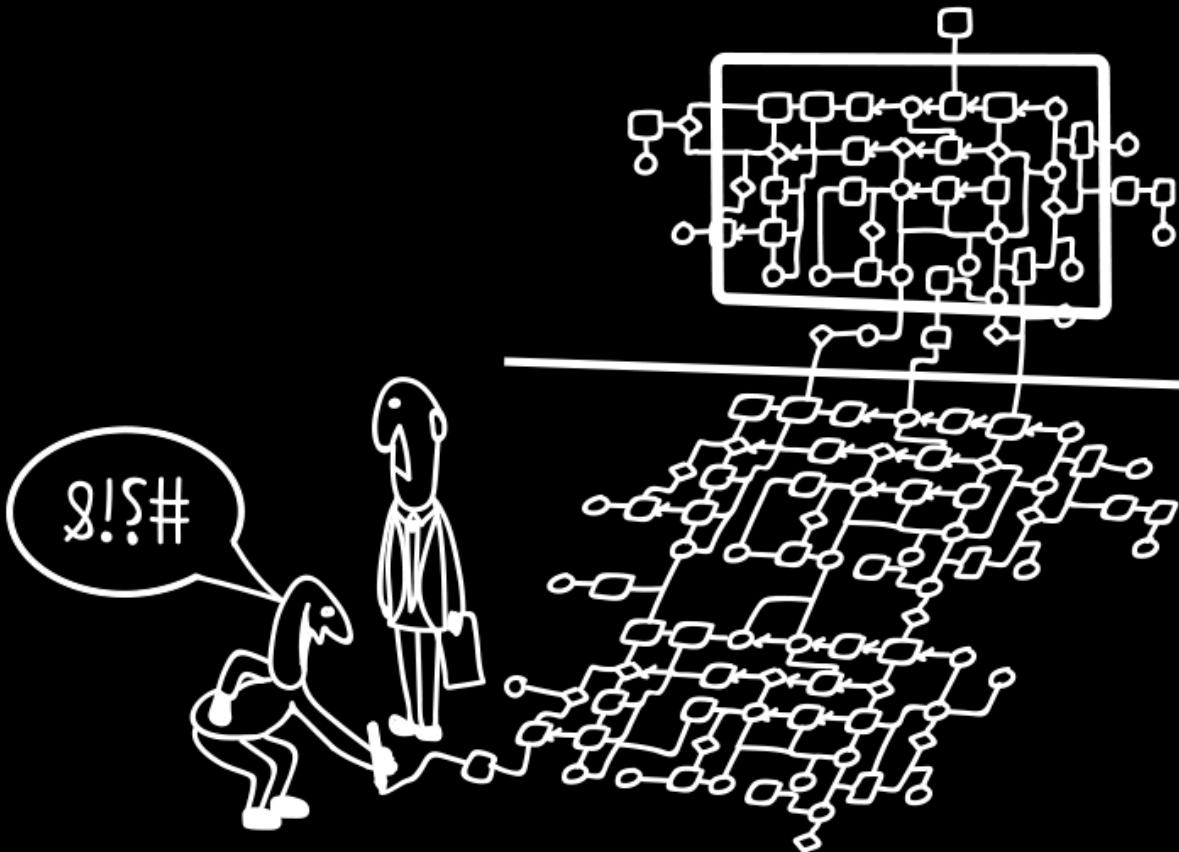
Orchestration



We have a new basic agreement
with the car rental agency and
can cancel for free within 1 hour
– do that first!

You have to adjust one service and redeploy only this one!

Graphical models?





Clemens Vasters

Architect at Microsoft

<http://vasters.com/archive/Sagas.html>

Clemens Vazquez
@vazquez_clemens
[Email](#)
[Twitter](#)
[Facebook](#)
[Google+](#)
[LinkedIn](#)
[XING](#)
[Instagram](#)
[Climate](#)
[DataScienceBerlin](#)
[Zurich](#)
[Nautilus](#)
[YouTube](#)
[Polar](#)
[Line](#)

Today has been a Italy day in some parts of the Timetables dealing with the Saga pattern. As it stands, there are a few requirements for "NT" users that uses the term "Saga" for zone framework implementation of a static deadline or workflow. Truth is that what is called a Saga is a Saga or a management pattern.

Saga come out of the realization that particularly long-haul transactions (originally open jazz interface), but also well-defined transactions across a location and/or trust boundaries can't easily be handled using the classic DCOM model with 2-phase commit and holding logic for the duration of the process because a Saga involves individual transaction whose effects can be, somehow, renounced after work has been performed and committed.



The glutton shows a single Saga. If you book a travel itinerary you want a car and a hotel and a flight. If you can't get all of them, it's probably not worth going. It's also very certain that you can't enjoy all of these pleasures in a discounted 40% transaction. Instead, you have an ability for booking rental cars that leaves both how to perform a reservation and also how to cancel it - and one for a hotel and one for a flight.

The athletes are grouped in a composite job (rotating dig) that's handled along the activity chain. If you want, you can dignify/energize the rotating dig teams so that they can only be understood and manipulated by the intended readers. When an activity completes, add a read of the congestion to the rotating dig along with information on where his compensating aggression can be readied (e.g. via a pause). When an activity fails, clean up locally and then conduct the rotating dig to determine the last congestion a client's compensation addressed to unwind the transaction outcome.

If you're a bit familiar with Java, you'll also notice that he organized the step by step reasoning a nonstop and always passed off what he had learned at the time, because the rail company can move more cars on one of those high demand. Reasoning a hand I'd slightly more rigid, but you can be extremely bodium of a reservation without penalty until 24 hours before the stay. Jay fare often comes with a refund notification, so you'll want to do that last.

I created a [Get on GitHub](#) that you can run as a console application. It illustrates the model in code. Mind that it's a reading and have a framework. I wrote this in less than 90 minutes, so don't expect to reuse

I have a [Git](#) on GitHub that you can run as a sample application. It illustrates this model in code. Mind that it is a module and not a framework; I wrote this in less than 90 minutes, so don't expect no reuse this.

The main program uses an example of a young girl (all the data are in the one file) and creates three completely independent "greenests" (adults/hosts) that are each responsible for handling a particular line of work. The "greenests" are linked by a "network" and each line of activity has an address for forward progress work and one of congegation work. The network resolution is illustrated by "Song".

```

  static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter a number between 1 and 100");
    int userNumber = scanner.nextInt();
    scanner.close();
    int randomNum = (int) (Math.random() * 100) + 1;
    System.out.println("I'm thinking of a number between 1 and 100");
    System.out.println("Is it " + randomNum + "?");
    String answer = scanner.nextLine();
    if (answer.equals("yes")) {
      System.out.println("I knew you would guess!");
    } else if (answer.equals("no")) {
      System.out.println("I'm sorry, I'm thinking of another number.");
    } else {
      System.out.println("I didn't understand your answer.");
    }
  }
}

```

Classmate
▼ Main
■ Home
■ Test
■ Past
■ Current
■ Live
■ Xmas
■ Help
■ Chat
■ Share
■ Groups
■ News
■ Photo
■ Video

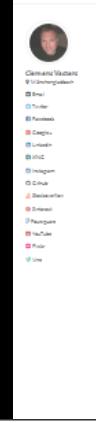
1

that it is and it's the only piece of information that flows between the collaborating entities. There is no annual synchronization. All work is local on the nodes and once a node is done, it then handles routing digi forward on its local or backbone (or failure). For forward progress data, the routing digi has a queue and the backbone does it in a tail-to-head. The routing digi also handles rescaling and load balancing whenever the "need" thing goes off on the way forward and backbone.

@berndruecker



Clemens Vasters
Architect at Microsoft



Saga
Today has been a busy day. In some parts of the Twinklance discussing the Saga pattern. In general, there are a few frameworks for NFT out there that use the term "Saga" for some framework implementation of a state machine or workflow. Trouble is, that's not what a Saga is. A Saga is a failure management pattern.

Saga come out of the realization that generally long-lived transactions (rightly so, since you can't update databases), but also for distributed transactional boundaries, can't easily be handled using the classic ACID model with 2-phase commit and holding locks for the duration of the work instead, a Saga splits work into individual transactions whose effects can be somehow, somehow after work has been performed and committed.



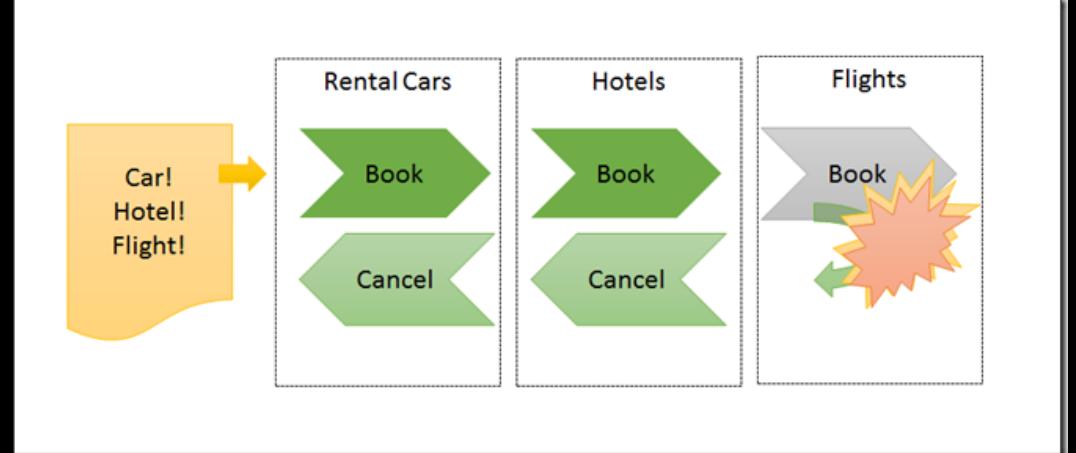
The picture shows a single Saga. If you want to travel, you want a car and a hotel and a flight. If you can't get all of them, it probably isn't worth going. It's also very certain that you can't end all of these processes in a atomized ACID transaction. Instead, you have an ability for booking rental cars that leaves both how to perform a reservation and also how to cancel it - and one for a hotel and one for flights.

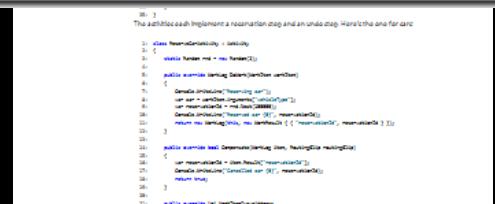
The activities are grouped in a composite job (routing job) that's handled along the activity chain. If you want, you can dignitize the routing job so that they can only be understood and manipulated by the intended reader. When an activity completes, it adds a record of its completion to the routing job along with information on where its consequential operation can be reached (e.g. it's obvious) when an activity fails, it discards itself and then picks the routing job back up to start the last

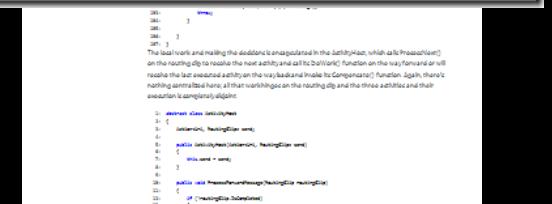


The sharding happened early through the routing job. The routing job is "translatable" (it's not pretend that it is) and it's the only place of information that flows between the collaborating entities. There is no central administration of work items, but the reading of the work items needs to done, to either handle the work item or to forward it to another entity. For example, if a car reservation fails, the routing job has a queue and for bad status it needs to make a state. The routing job also handles resuming whatever the "head" thing to do is on the way forward and so on.









<http://vasters.com/archive/Sagas.html>



Clemens Vasters
Architect at Microsoft

<http://vasters.com/archive/Sagas.html>

The picture shows one of the illustrations that gamifies long-lived transaction (originally from Jim Webber's database), but also illustrates transactions and/or must boundaries can't easily be handled using the classic ACID model with 2-phase commit and holding locks for the duration of the work instead. A Saga splits work into individual transactions whose effects can be somehow, recombined after work has been performed and committed.

The picture shows a single Saga. If you like to travel, you want a car and a hotel and a flight. If you can't get all of them, it's probably not worth going. It's also very certain that you can't end all of these processes in a simultaneous ACID transaction. Instead, you have an ability for booking rental cars that leaves both here to perform a reservation and also have an ability – one for a hotel and one for flights.

The activities are grouped in a composite job (reading db) that's handled along the activity chain. If you want, you can diagrammize the reading db stage so that they can only be understood and manipulated by the intended reader. When an activity completes, it adds a record of the completion to the reading db along with information on where its corresponding operation is to be used (e.g. via a queue). When an activity fails, it does cleanup and then sends the reading db back to the last completed activity's completion to resume the next iteration scheme.

If you're a bit more experienced with distributed systems, you might have heard about the concept of a saga manager. If you keep the saga manager in the same process, it's much easier to manage more and more of them in high demand. Reserving a hotel, flight, more flights, but you can even complete a reservation without penalty until 24h before the stay. Before often comes with a refund reservation, so you have to do that.

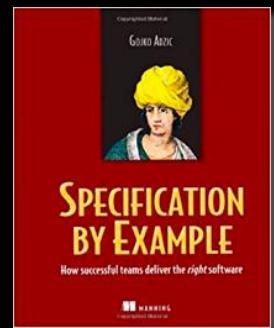
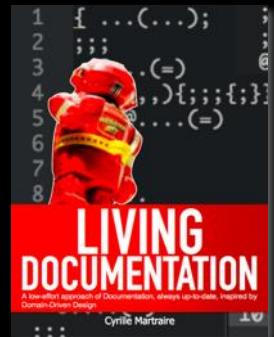
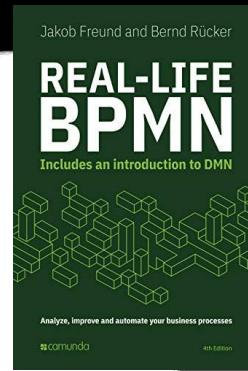
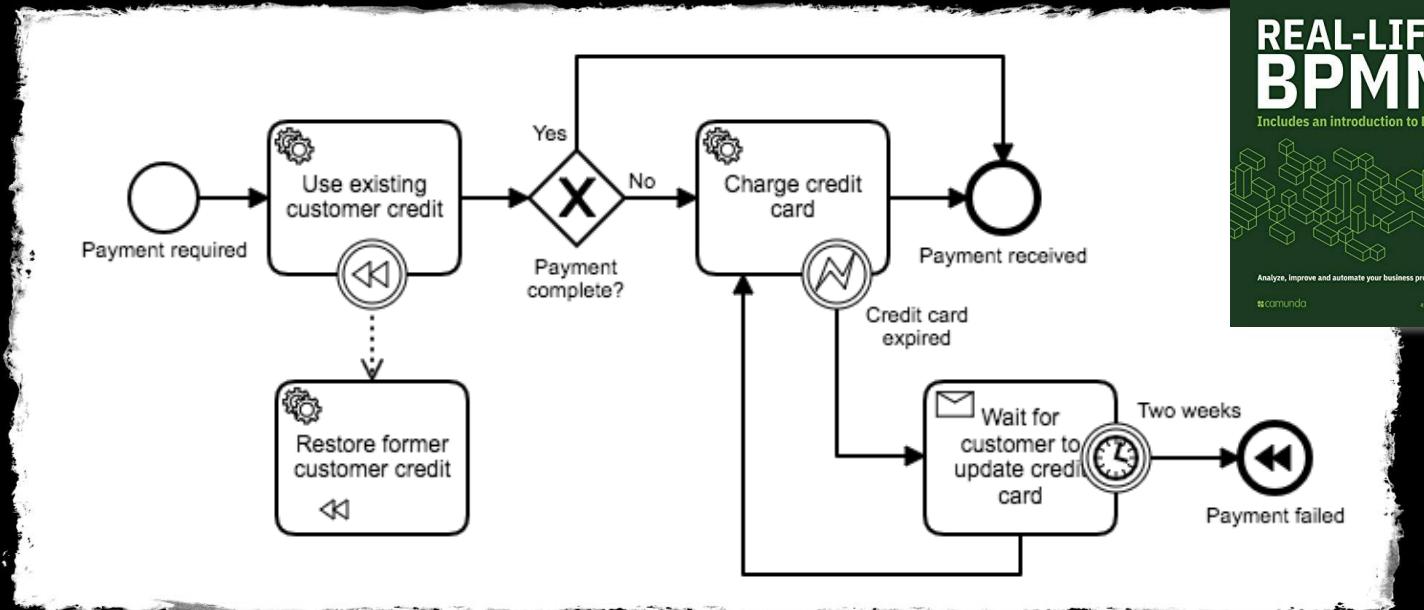
I learned a lot on GitHub that can run as a sample application. It illustrates this model in code. What that is to say is that it's a module and not a framework. You'd like to less than 90 minutes, so don't expect to reuse this.

The main program starts an example reading db that's database in the use field and creates three concurrent independent "processors" (activity hand) that are each responsible for handling a particular line of work. The "processors" are linked by "previews" and each line of activity has an associated forward progress and one of some organization tasks. The work prioritization is distributed by "Sends".

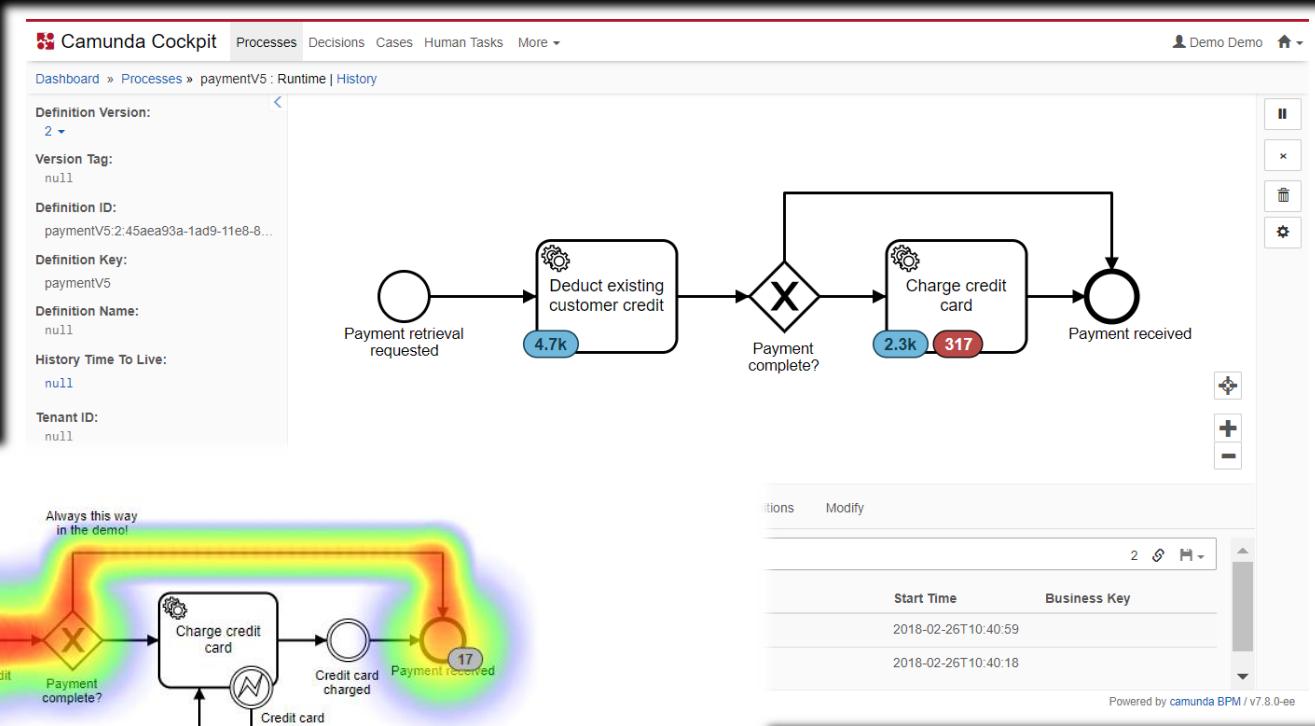
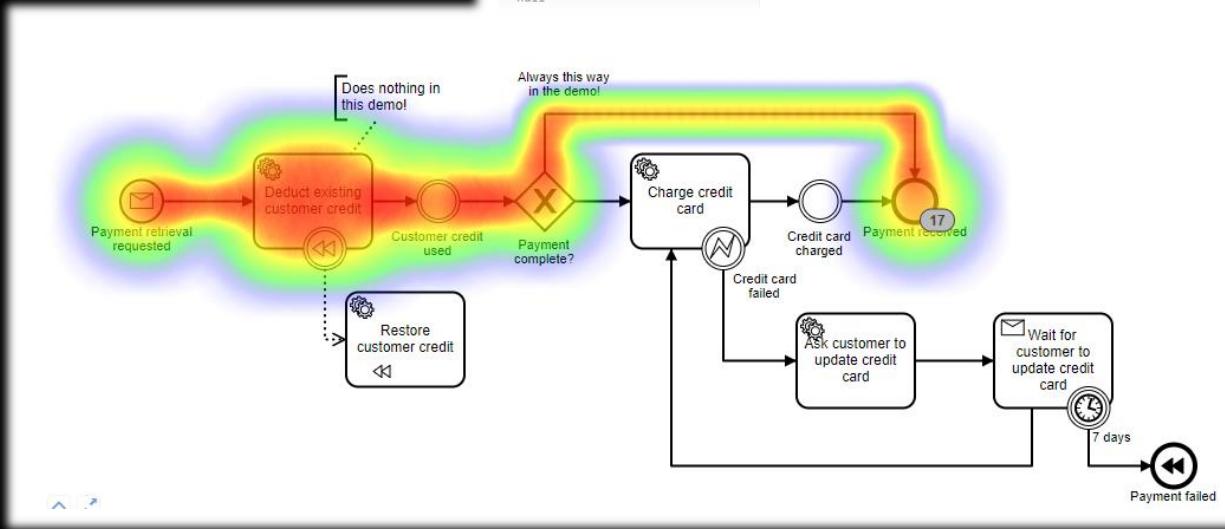
```

1: static ActivityPreview[] previews;
2:
3: static void main(String[] args) {
4:     ...
5:     var readingDb = new ReadingDb();
6:     ...
7:     var processor1 = new ReadingDbProcessor(readingDb);
8:     var processor2 = new ReadingDbProcessor(readingDb);
9:     var processor3 = new ReadingDbProcessor(readingDb);
10:
11:
12:    // begin three being sequentially previewed processor with status between them
13:    previews = new ActivityPreview[3];
14:    ...
15:    ...
16:    ...
17:    ...
18:    ...
19:    ...
20:    ...
21:    ...
22:    ...
23:    ...
24:    ...
25:    ...
26:    ...
27:    ...
28:    ...
29:    ...
30:    ...
31:    ...
32:    ...
33:    ...
34:    ...
35:    ...
36:    ...
37:    ...
38:    ...
39:    ...
40:    ...
41:    ...
42:    ...
43:    ...
44:    ...
45:    ...
46:    ...
47:    ...
48:    ...
49:    ...
50:    ...
51:    ...
52:    ...
53:    ...
54:    ...
55:    ...
56:    ...
57:    ...
58:    ...
59:    ...
60:    ...
61:    ...
62:    ...
63:    ...
64:    ...
65:    ...
66:    ...
67:    ...
68:    ...
69:    ...
70:    ...
71:    ...
72:    ...
73:    ...
74:    ...
75:    ...
76:    ...
77:    ...
78:    ...
79:    ...
80:    ...
81:    ...
82:    ...
83:    ...
84:    ...
85:    ...
86:    ...
87:    ...
88:    ...
89:    ...
90:    ...
91:    ...
92:    ...
93:    ...
94:    ...
95:    ...
96:    ...
97:    ...
98:    ...
99:    ...
100:   ...
101:   ...
102:   ...
103:   ...
104:   ...
105:   ...
106:   ...
107:   ...
108:   ...
109:   ...
110:   ...
111:   ...
112:   ...
113:   ...
114:   ...
115:   ...
116:   ...
117:   ...
118:   ...
119:   ...
120:   ...
121:   ...
122:   ...
123:   ...
124:   ...
125:   ...
126:   ...
127:   ...
128:   ...
129:   ...
130:   ...
131:   ...
132:   ...
133:   ...
134:   ...
135:   ...
136:   ...
137:   ...
138:   ...
139:   ...
140:   ...
141:   ...
142:   ...
143:   ...
144:   ...
145:   ...
146:   ...
147:   ...
148:   ...
149:   ...
150:   ...
151:   ...
152:   ...
153:   ...
154:   ...
155:   ...
156:   ...
157:   ...
158:   ...
159:   ...
160:   ...
161:   ...
162:   ...
163:   ...
164:   ...
165:   ...
166:   ...
167:   ...
168:   ...
169:   ...
170:   ...
171:   ...
172:   ...
173:   ...
174:   ...
175:   ...
176:   ...
177:   ...
178:   ...
179:   ...
180:   ...
181:   ...
182:   ...
183:   ...
184:   ...
185:   ...
186:   ...
187:   ...
188:   ...
189:   ...
190:   ...
191:   ...
192:   ...
193:   ...
194:   ...
195:   ...
196:   ...
197:   ...
198:   ...
199:   ...
200:   ...
201:   ...
202:   ...
203:   ...
204:   ...
205:   ...
206:   ...
207:   ...
208:   ...
209:   ...
210:   ...
211:   ...
212:   ...
213:   ...
214:   ...
215:   ...
216:   ...
217:   ...
218:   ...
219:   ...
220:   ...
221:   ...
222:   ...
223:   ...
224:   ...
225:   ...
226:   ...
227:   ...
228:   ...
229:   ...
230:   ...
231:   ...
232:   ...
233:   ...
234:   ...
235:   ...
236:   ...
237:   ...
238:   ...
239:   ...
240:   ...
241:   ...
242:   ...
243:   ...
244:   ...
245:   ...
246:   ...
247:   ...
248:   ...
249:   ...
250:   ...
251:   ...
252:   ...
253:   ...
254:   ...
255:   ...
256:   ...
257:   ...
258:   ...
259:   ...
260:   ...
261:   ...
262:   ...
263:   ...
264:   ...
265:   ...
266:   ...
267:   ...
268:   ...
269:   ...
270:   ...
271:   ...
272:   ...
273:   ...
274:   ...
275:   ...
276:   ...
277:   ...
278:   ...
279:   ...
280:   ...
281:   ...
282:   ...
283:   ...
284:   ...
285:   ...
286:   ...
287:   ...
288:   ...
289:   ...
290:   ...
291:   ...
292:   ...
293:   ...
294:   ...
295:   ...
296:   ...
297:   ...
298:   ...
299:   ...
300:   ...
301:   ...
302:   ...
303:   ...
304:   ...
305:   ...
306:   ...
307:   ...
308:   ...
309:   ...
310:   ...
311:   ...
312:   ...
313:   ...
314:   ...
315:   ...
316:   ...
317:   ...
318:   ...
319:   ...
320:   ...
321:   ...
322:   ...
323:   ...
324:   ...
325:   ...
326:   ...
327:   ...
328:   ...
329:   ...
330:   ...
331:   ...
332:   ...
333:   ...
334:   ...
335:   ...
336:   ...
337:   ...
338:   ...
339:   ...
340:   ...
341:   ...
342:   ...
343:   ...
344:   ...
345:   ...
346:   ...
347:   ...
348:   ...
349:   ...
350:   ...
351:   ...
352:   ...
353:   ...
354:   ...
355:   ...
356:   ...
357:   ...
358:   ...
359:   ...
360:   ...
361:   ...
362:   ...
363:   ...
364:   ...
365:   ...
366:   ...
367:   ...
368:   ...
369:   ...
370:   ...
371:   ...
372:   ...
373:   ...
374:   ...
375:   ...
376:   ...
377:   ...
378:   ...
379:   ...
380:   ...
381:   ...
382:   ...
383:   ...
384:   ...
385:   ...
386:   ...
387:   ...
388:   ...
389:   ...
390:   ...
391:   ...
392:   ...
393:   ...
394:   ...
395:   ...
396:   ...
397:   ...
398:   ...
399:   ...
400:   ...
401:   ...
402:   ...
403:   ...
404:   ...
405:   ...
406:   ...
407:   ...
408:   ...
409:   ...
410:   ...
411:   ...
412:   ...
413:   ...
414:   ...
415:   ...
416:   ...
417:   ...
418:   ...
419:   ...
420:   ...
421:   ...
422:   ...
423:   ...
424:   ...
425:   ...
426:   ...
427:   ...
428:   ...
429:   ...
430:   ...
431:   ...
432:   ...
433:   ...
434:   ...
435:   ...
436:   ...
437:   ...
438:   ...
439:   ...
440:   ...
441:   ...
442:   ...
443:   ...
444:   ...
445:   ...
446:   ...
447:   ...
448:   ...
449:   ...
450:   ...
451:   ...
452:   ...
453:   ...
454:   ...
455:   ...
456:   ...
457:   ...
458:   ...
459:   ...
460:   ...
461:   ...
462:   ...
463:   ...
464:   ...
465:   ...
466:   ...
467:   ...
468:   ...
469:   ...
470:   ...
471:   ...
472:   ...
473:   ...
474:   ...
475:   ...
476:   ...
477:   ...
478:   ...
479:   ...
480:   ...
481:   ...
482:   ...
483:   ...
484:   ...
485:   ...
486:   ...
487:   ...
488:   ...
489:   ...
490:   ...
491:   ...
492:   ...
493:   ...
494:   ...
495:   ...
496:   ...
497:   ...
498:   ...
499:   ...
500:   ...
501:   ...
502:   ...
503:   ...
504:   ...
505:   ...
506:   ...
507:   ...
508:   ...
509:   ...
510:   ...
511:   ...
512:   ...
513:   ...
514:   ...
515:   ...
516:   ...
517:   ...
518:   ...
519:   ...
520:   ...
521:   ...
522:   ...
523:   ...
524:   ...
525:   ...
526:   ...
527:   ...
528:   ...
529:   ...
530:   ...
531:   ...
532:   ...
533:   ...
534:   ...
535:   ...
536:   ...
537:   ...
538:   ...
539:   ...
540:   ...
541:   ...
542:   ...
543:   ...
544:   ...
545:   ...
546:   ...
547:   ...
548:   ...
549:   ...
550:   ...
551:   ...
552:   ...
553:   ...
554:   ...
555:   ...
556:   ...
557:   ...
558:   ...
559:   ...
560:   ...
561:   ...
562:   ...
563:   ...
564:   ...
565:   ...
566:   ...
567:   ...
568:   ...
569:   ...
570:   ...
571:   ...
572:   ...
573:   ...
574:   ...
575:   ...
576:   ...
577:   ...
578:   ...
579:   ...
580:   ...
581:   ...
582:   ...
583:   ...
584:   ...
585:   ...
586:   ...
587:   ...
588:   ...
589:   ...
590:   ...
591:   ...
592:   ...
593:   ...
594:   ...
595:   ...
596:   ...
597:   ...
598:   ...
599:   ...
600:   ...
601:   ...
602:   ...
603:   ...
604:   ...
605:   ...
606:   ...
607:   ...
608:   ...
609:   ...
610:   ...
611:   ...
612:   ...
613:   ...
614:   ...
615:   ...
616:   ...
617:   ...
618:   ...
619:   ...
620:   ...
621:   ...
622:   ...
623:   ...
624:   ...
625:   ...
626:   ...
627:   ...
628:   ...
629:   ...
630:   ...
631:   ...
632:   ...
633:   ...
634:   ...
635:   ...
636:   ...
637:   ...
638:   ...
639:   ...
640:   ...
641:   ...
642:   ...
643:   ...
644:   ...
645:   ...
646:   ...
647:   ...
648:   ...
649:   ...
650:   ...
651:   ...
652:   ...
653:   ...
654:   ...
655:   ...
656:   ...
657:   ...
658:   ...
659:   ...
660:   ...
661:   ...
662:   ...
663:   ...
664:   ...
665:   ...
666:   ...
667:   ...
668:   ...
669:   ...
670:   ...
671:   ...
672:   ...
673:   ...
674:   ...
675:   ...
676:   ...
677:   ...
678:   ...
679:   ...
680:   ...
681:   ...
682:   ...
683:   ...
684:   ...
685:   ...
686:   ...
687:   ...
688:   ...
689:   ...
690:   ...
691:   ...
692:   ...
693:   ...
694:   ...
695:   ...
696:   ...
697:   ...
698:   ...
699:   ...
700:   ...
701:   ...
702:   ...
703:   ...
704:   ...
705:   ...
706:   ...
707:   ...
708:   ...
709:   ...
710:   ...
711:   ...
712:   ...
713:   ...
714:   ...
715:   ...
716:   ...
717:   ...
718:   ...
719:   ...
720:   ...
721:   ...
722:   ...
723:   ...
724:   ...
725:   ...
726:   ...
727:   ...
728:   ...
729:   ...
730:   ...
731:   ...
732:   ...
733:   ...
734:   ...
735:   ...
736:   ...
737:   ...
738:   ...
739:   ...
740:   ...
741:   ...
742:   ...
743:   ...
744:   ...
745:   ...
746:   ...
747:   ...
748:   ...
749:   ...
750:   ...
751:   ...
752:   ...
753:   ...
754:   ...
755:   ...
756:   ...
757:   ...
758:   ...
759:   ...
760:   ...
761:   ...
762:   ...
763:   ...
764:   ...
765:   ...
766:   ...
767:   ...
768:   ...
769:   ...
770:   ...
771:   ...
772:   ...
773:   ...
774:   ...
775:   ...
776:   ...
777:   ...
778:   ...
779:   ...
780:   ...
781:   ...
782:   ...
783:   ...
784:   ...
785:   ...
786:   ...
787:   ...
788:   ...
789:   ...
790:   ...
791:   ...
792:   ...
793:   ...
794:   ...
795:   ...
796:   ...
797:   ...
798:   ...
799:   ...
800:   ...
801:   ...
802:   ...
803:   ...
804:   ...
805:   ...
806:   ...
807:   ...
808:   ...
809:   ...
810:   ...
811:   ...
812:   ...
813:   ...
814:   ...
815:   ...
816:   ...
817:   ...
818:   ...
819:   ...
820:   ...
821:   ...
822:   ...
823:   ...
824:   ...
825:   ...
826:   ...
827:   ...
828:   ...
829:   ...
830:   ...
831:   ...
832:   ...
833:   ...
834:   ...
835:   ...
836:   ...
837:   ...
838:   ...
839:   ...
840:   ...
841:   ...
842:   ...
843:   ...
844:   ...
845:   ...
846:   ...
847:   ...
848:   ...
849:   ...
850:   ...
851:   ...
852:   ...
853:   ...
854:   ...
855:   ...
856:   ...
857:   ...
858:   ...
859:   ...
860:   ...
861:   ...
862:   ...
863:   ...
864:   ...
865:   ...
866:   ...
867:   ...
868:   ...
869:   ...
870:   ...
871:   ...
872:   ...
873:   ...
874:   ...
875:   ...
876:   ...
877:   ...
878:   ...
879:   ...
880:   ...
881:   ...
882:   ...
883:   ...
884:   ...
885:   ...
886:   ...
887:   ...
888:   ...
889:   ...
890:   ...
891:   ...
892:   ...
893:   ...
894:   ...
895:   ...
896:   ...
897:   ...
898:   ...
899:   ...
900:   ...
901:   ...
902:   ...
903:   ...
904:   ...
905:   ...
906:   ...
907:   ...
908:   ...
909:   ...
910:   ...
911:   ...
912:   ...
913:   ...
914:   ...
915:   ...
916:   ...
917:   ...
918:   ...
919:   ...
920:   ...
921:   ...
922:   ...
923:   ...
924:   ...
925:   ...
926:   ...
927:   ...
928:   ...
929:   ...
930:   ...
931:   ...
932:   ...
933:   ...
934:   ...
935:   ...
936:   ...
937:   ...
938:   ...
939:   ...
940:   ...
941:   ...
942:   ...
943:   ...
944:   ...
945:   ...
946:   ...
947:   ...
948:   ...
949:   ...
950:   ...
951:   ...
952:   ...
953:   ...
954:   ...
955:   ...
956:   ...
957:   ...
958:   ...
959:   ...
960:   ...
961:   ...
962:   ...
963:   ...
964:   ...
965:   ...
966:   ...
967:   ...
968:   ...
969:   ...
970:   ...
971:   ...
972:   ...
973:   ...
974:   ...
975:   ...
976:   ...
977:   ...
978:   ...
979:   ...
980:   ...
981:   ...
982:   ...
983:   ...
984:   ...
985:   ...
986:   ...
987:   ...
988:   ...
989:   ...
990:   ...
991:   ...
992:   ...
993:   ...
994:   ...
995:   ...
996:   ...
997:   ...
998:   ...
999:   ...
1000:   ...
1001:   ...
1002:   ...
1003:   ...
1004:   ...
1005:   ...
1006:   ...
1007:   ...
1008:   ...
1009:   ...
1010:   ...
1011:   ...
1012:   ...
1013:   ...
1014:   ...
1015:   ...
1016:   ...
1017:   ...
1018:   ...
1019:   ...
1020:   ...
1021:   ...
1022:   ...
1023:   ...
1024:   ...
1025:   ...
1026:   ...
1027:   ...
1028:   ...
1029:   ...
1030:   ...
1031:   ...
1032:   ...
1033:   ...
1034:   ...
1035:   ...
1036:   ...
1037:   ...
1038:   ...
1039:   ...
1040:   ...
1041:   ...
1042:   ...
1043:   ...
1044:   ...
1045:   ...
1046:   ...
1047:   ...
1048:   ...
1049:   ...
1050:   ...
1051:   ...
1052:   ...
1053:   ...
1054:   ...
1055:   ...
1056:   ...
1057:   ...
1058:   ...
1059:   ...
1060:   ...
1061:   ...
1062:   ...
1063:   ...
1064:   ...
1065:   ...
1066:   ...
1067:   ...
1068:   ...
1069:   ...
1070:   ...
1071:   ...
1072:   ...
1073:   ...
1074:   ...
1075:   ...
1076:   ...
1077:   ...
1078:   ...
1079:   ...
1080:   ...
1081:   ...
1082:   ...
1083:   ...
1084:   ...
1085:   ...
1086:   ...
1087:   ...
1088:   ...
1089:   ...
1090:   ...
1091:   ...
1092:   ...
1093:   ...
1094:   ...
1095:   ...
1096:   ...
1097:   ...
1098:   ...
1099:   ...
1100:   ...
1101:   ...
1102:   ...
1103:   ...
1104:   ...
1105:   ...
1106:   ...
1107:   ...
1108:   ...
1109:   ...
1110:   ...
1111:   ...
1112:   ...
1113:   ...
1114:   ...
1115:   ...
1116:   ...
1117:   ...
1118:   ...
1119:   ...
1120:   ...
1121:   ...
1122:   ...
1123:   ...
1124:   ...
1125:   ...
1126:   ...
1127:   ...
1128:   ...
1129:   ...
1130:   ...
1131:   ...
1132:   ...
1133:   ...
1134:   ...
1135:   ...
1136:   ...
1137:   ...
1138:   ...
1139:   ...
1140:   ...
1141:   ...
1142:   ...
1143:   ...
1144:   ...
1145:   ...
1146:   ...
1147:   ...
1148:   ...
1149:   ...
1150:   ...
1151:   ...
1152:   ...
1153:   ...
1154:   ...
1155:   ...
1156:   ...
1157:   ...
1158:   ...
1159:   ...
1160:   ...
1161:   ...
1162:   ...
1163:   ...
1164:   ...
1165:   ...
1166:   ...
1167:   ...
1168:   ...
1169:   ...
1170:   ...
1171:   ...
1172:   ...
1173:   ...
1174:   ...
1175:   ...
1176:   ...
1177:   ...
1178:   ...
1179:   ...
1180:   ...
1181:   ...
1182:   ...
1183:   ...
1184:   ...
1185:   ...
1186:   ...
1187:   ...
1188:   ...
1189:   ...
1190:   ...
1191:   ...
1192:   ...
1193:   ...
1194:   ...
1195:   ...
1196:   ...
1197:   ...
1198:   ...
1199:   ...
1200:   ...
1201:   ...
1202:   ...
1203:   ...
1204:   ...
1205:   ...
1206:   ...
1207:   ...
1208:   ...
1209:   ...
1210:   ...
1211:   ...
1212:   ...
1213:   ...
1214:   ...
1215:   ...
1216:   ...
1217:   ...
1218:   ...
1219:   ...
1220:   ...
1221:   ...
1222:   ...
1223:   ...
1224:   ...
1225:   ...
1226:   ...
1227:   ...
1228:   ...
1229:   ...
1230:   ...
1231:   ...
1232:   ...
1233:   ...
1234:   ...
1235:   ...
1236:   ...
1237:   ...
1238:   ...
1239:   ...
1240:   ...
1241:   ...
1242:   ...
1243:   ...
1244:   ...
1245:   ...
1246:   ...
1247:   ...
1248:   ...
1249:   ...
1250:   ...
1251:   ...
1252:   ...
1253:   ...
1254:   ...
1255:   ...
1256:   ...
1257:   ...
1258:   ...
1259:   ...
1260:   ...
1261:   ...
1262:   ...
1263:   ...
1264:   ...
1265:   ...
1266:   ...
1267:   ...
1268:   ...
1269:   ...
1270:   ...
1271:   ...
1272:   ...
1273:   ...
1274:   ...
1275:   ...
1276:   ...
1277:   ...
1278:   ...
1279:   ...
1280:   ...
1281:   ...
1282:   ...
1283:   ...
1284:   ...
1285:   ...
1286:   ...
1287:   ...
1288:   ...
1289:   ...
1290:   ...
1291:   ...
1292:   ...
1293:   ...
1294:   ...
1295:   ...
1296:   ...
1297:   ...
1298:   ...
1299:   ...
1300:   ...
1301:   ...
1302:   ...
1303:   ...
1304:   ...
1305:   ...
1306:   ...
1307:   ...
1308:   ...
1309:   ...
1310:   ...
1311:   ...
1312:   ...
1313:   ...
1314:   ...
1315:   ...
1316:   ...
1317:   ...
1318:   ...
1319:   ...
1320:   ...
1321:   ...
1322:   ...
1323:   ...
1324:   ...
1325:   ...
1326:   ...
1327:   ...
1328:   ...
1329:   ...
1330:   ...
1331:   ...
1332:   ...
1333:   ...
1334:   ...
1335:   ...
1336:   ...
1337:   ...
1338:   ...
1339:   ...
1340:   ...
1341:   ...
1342:   ...
1343:   ...
1344:   ...
1345:   ...
1346:   ...
1347:   ...
1348:   ...
1349:   ...
1350:   ...
1351:   ...
1352:   ...
1353:   ...
1354:   ...
1355:   ...
1356:   ...
1357:   ...
1358:   ...
1359:   ...
1360:   ...
1361:   ...
1362:   ...
1363:   ...
1364:   ...
1365:   ...
1366:   ...
1367:   ...
1368:   ...
1369:   ...
1370:   ...
1371:   ...
1372:   ...
1373:   ...
1374:   ...
1375:   ...
1376:   ...
1377:   ...
1378:   ...
1379:   ...
1380:   ...
1381:   ...
1382:   ...
1383:   ...
1384:   ...
1385:   ...
1386:   ...
1387:   ...
1388:   ...
1389:   ...
1390:   ...
1391:   ...
1392:   ...
1393:   ...
1394:   ...
1395:   ...
1396:   ...
1397:   ...
1398:   ...
1399:   ...
1400:   ...
1401:   ...
1402:   ...
1403:   ...
1404:   ...
1405:   ...
1406:   ...
1407:   ...
1408:   ...
1409:   ...
1410:   ...
1411:   ...
1412:   ...
1413:   ...
1414:   ...
1415:   ...
1416:   ...
1417:   ...
1418:   ...
1419:   ...
1420:   ...
1421:   ...
1422:   ...
1423:   ...
1424:   ...
1425:   ...
1426:   ...
1427:   ...
1428:   ...
1429:   ...
1430:   ...
1431:   ...
1432:   ...
1433:   ...
1434:   ...
1435:   ...
1436:   ...
1437:   ...
1438:   ...
1439:   ...
1440:   ...
1441:   ...
1442:   ...
1443:   ...
1444:   ...
1445:   ...
1446:   ...
1447:   ...
1448:   ...
1449:   ...
1450:   ...
1451:   ...
1452:   ...
1453:   ...
1454:   ...
1455:   ...
1456:   ...
1457:   ...
1458:   ...
1459:   ...
1460:   ...
1461:   ...
1462:   ...
1463:   ...
1464:   ...
1465:   ...
1466:   ...
1467:   ...
1468:   ...
1469:   ...
1470:   ...
1471:   ...
1472:   ...
1473:   ...
1474:   ...
1475:   ...
1476:   ...
1477:   ...
1478:   ...
1479:   ...
1480:   ...
1481:   ...
1482:   ...
1483:   ...
1484:   ...
1485:   ...
1486:   ...
1487:   ...
1488:   ...
1489:   ...
1490:   ...
1491:   ...
1492:   ...
1493:   ...
1494:   ...
1495:   ...
1496:   ...
1497:   ...
1498:   ...
1499:   ...
1500:   ...
1501:   ...
1502:   ...
1503:   ...
1504:   ...
1505:   ...
1506:   ...
1507:   ...
1508:   ...
1509:   ...
1510:   ...
1511:   ...
1512:   ...
1513:   ...
1514:   ...
1515:   ...
1516:   ...
1517:   ...
1518:   ...
1519:   ...
1520:   ...
1521:   ...
1522:   ...
1523:   ...
1524:   ...
1525:   ...
1526:   ...
1527:   ...
1528:   ...
1529:   ...
1530:   ...
1531:   ...
1532:   ...
1533:   ...
1534:   ...
1535:   ...
1536:   ...
1537:   ...
1538:   ...
1539:   ...
1540:   ...
1541:   ...
1542:   ...
1543:   ...
1544:   ...
1545:   ...
1546:   ...
1547:   ...
1548:   ...
1549:   ...
1550:   ...
1551:   ...
1552:   ...
1553:   ...
1554:   ...
1555:   ...
1556:   ...
1557:   ...
1558:   ...
1559:   ...
1560:   ...
1561:   ...
1562:   ...
1563:   ...
1564:   ...
1565:   ...
1566:   ...
1567:   ...
1568:   ...
1569:   ...
1570:   ...
1571:   ...
1572:   ...
1573:   ...
1574:   ...
1575:   ...
1576:   ...
1577:   ...
1578:   ...
1579:   ...
1580:   ...
1581:   ...
1582:   ...
1583:   ...
1584:   ...
1585:   ...
1586:   ...
1587:   ...
1588:   ...
1589:   ...
1590:   ...
1591:   ...
1592:   ...
1593:   ...
1594:   ...
1595:   ...
1596:   ...
1597:   ...
1598:   ...
1599:   ...
1600:   ...
1601:   ...
1602:   ...
1603:   ...
1604:   ...
1605:   ...
1606:   ...
1607:   ...
1608:   ...
1609:   ...
1610:   ...
1611:   ...
1612:   ...
1613:   ...
1614:   ...
1615:   ...
1616:   ...
1617:   ...
1618:   ...
1619:   ...
1620:   ...
1621:   ...
1622:   ...
1623:   ...
1624:   ...
1625:   ...
1626:   ...
1627:   ...
1628:   ...
1629:   ...
1630:   ...
1631:   ...
1632:   ...
1633:   ...
1634:   ...
1635:   ...
1636:   ...
1637:   ...
1638:   ...
1639:   ...
1640:   ...
1641:   ...
1642:   ...
1643:   ...
1644:   ...
1645:   ...
1646:   ...
1647:   ...
1648:   ...
1649:   ...
1650:   ...
1651:   ...
1652:   ...
1653:   ...
1654:   ...
1655:   ...
1656:   ...
1657:   ...
1658:   ...
1659:   ...
1660:   ...
1661:   ...
1662:   ...
1663:   ...
1664:   ...
1665:   ...
1666:   ...
1667:   ...
1668:   ...
1669:   ...
1670:   ...
1671:   ...
1672:   ...
1673:   ...
1674:   ...
1675:   ...
1676:   ...
1677:   ...
1678:   ...
1679:   ...
1680:   ...
1681:   ...
1682:   ...
1683:   ...
1684:   ...
1685:   ...
1686:   ...
1687:   ...
1688:   ...
1689:   ...
1690:   ...
1691:   ...
1692:   ...
1693:   ...
1694:   ...
1695:   ...
1696:   ...
1697:   ...
1698:   ...
1699:   ...
1700:   ...
1701:   ...
1702:   ...
1703:   ...
1704:   ...
1705:   ...
1706:   ...
1707:   ...
1708:   ...
1709:   ...
1710:   ...
1711:   ...
1712:   ...
1713:   ...
1714:   ...
1715:   ...
1716:   ...
1717:   ...
1718:   ...
1719:   ...
1720:   ...
1721:   ...
1722:   ...
1723:   ...
1724:   ...
1725:   ...
1726:   ...
1727:   ...
1728:   ...
1729:   ...
1730:   ...
1731:   ...
1732:   ...
1733:   ...
1734:   ...
1735:   ...
1736:   ...
1737:   ...
1738:   ...
1739:   ...
1740:   ...
1741:   ...
1742:   ...
1743:   ...
1744:   ...
1745:   ...
1746:   ...
1747:   ...
1748:   ...
1749:   ...
1750:   ...
1751:   ...
1752:   ...
1753:   ...
1754:   ...
1755:   ...
1756:   ...
1757:   ...
1758:   ...
1759:   ...
1760:   ...
1761:   ...
1762:   ...
1763:   ...
1764:   ...
1765:   ...
1766:   ...
1767:   ...
1768:   ...
1769:   ...
1770:   ...
1771:   ...
1772:   ...
1773:   ...
1774:   ...
1775:   ...
17
```

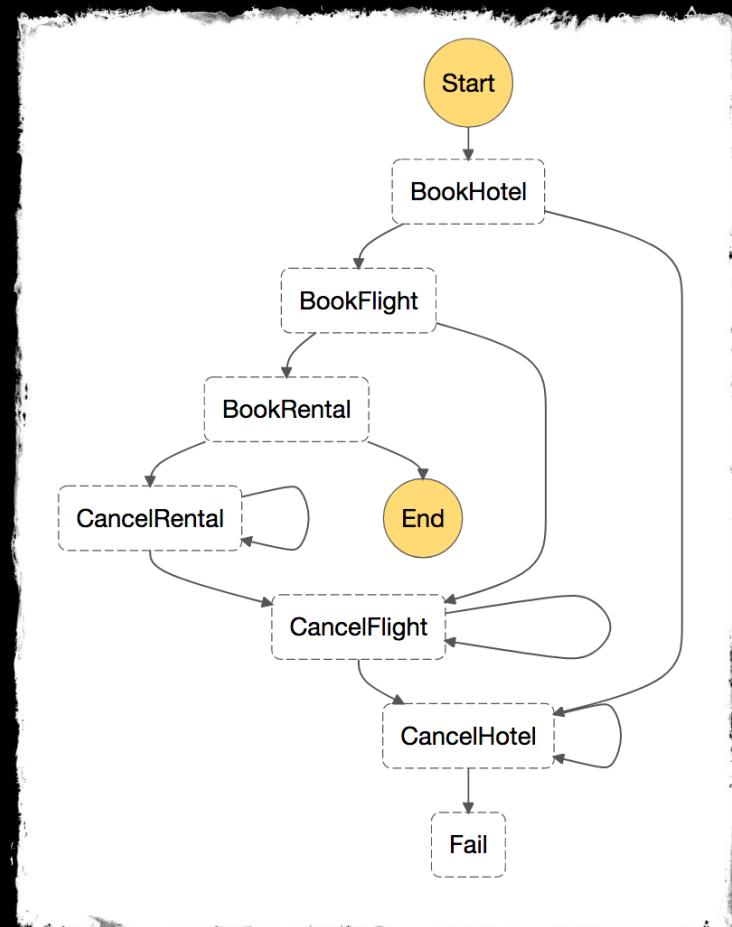
Living documentation for long-running behaviour



BizDevops

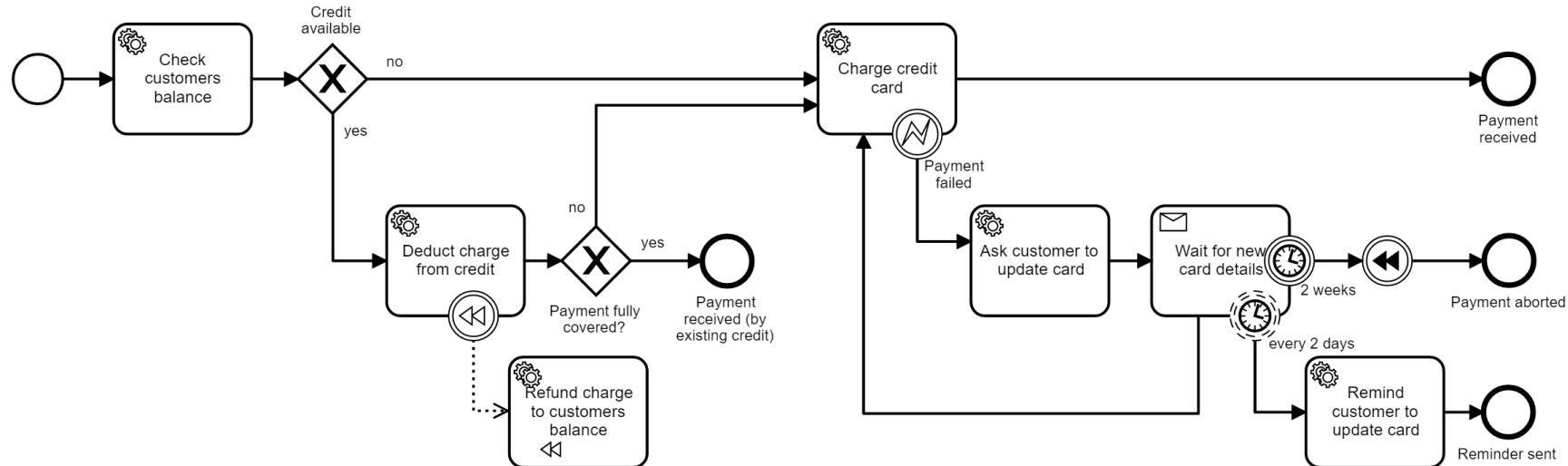


Step Functions



<https://read.acloud.guru/how-the-saga-pattern-manages-failures-with-aws-lambda-and-step-functions-bc8f7129f900>

Think of more complicated scenarios...



Recap

- Function coordination might include orchestration and workflow
- Carefully decide about choreography vs. orchestration
- Learn to handle eventual consistency and related business problems (e.g. transactions), e.g. by the Saga Pattern
- Idempotency is super important!

Thank you!



Contact: mail@berndruecker.io
@berndruecker

Slides: <https://berndruecker.io>

Blog: <https://medium.com/berndruecker>

Code: <https://github.com/berndruecker>



[https://www.infoworld.com/article/3254777/
application-development/
3-common-pitfalls-of-microservices-
integrationand-how-to-avoid-them.html](https://www.infoworld.com/article/3254777/application-development/3-common-pitfalls-of-microservices-integrationand-how-to-avoid-them.html)



[https://www.infoq.com/articles/events-
workflow-automation](https://www.infoq.com/articles/events-workflow-automation)



[https://thenewstack.io/5-workflow-automation-
use-cases-you-might-not-have-considered/](https://thenewstack.io/5-workflow-automation-use-cases-you-might-not-have-considered/)

